



**Universitat Autònoma
de Barcelona**

DETECCIÓ DE MATRÍCULES AMB SISTEMES DE BAIX COST

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes
realitzat per

Alberto Valero Moreno

i dirigit per

Ramon Baldrich

Escola d'Enginyeria

Sabadell, Setembre de 2012

El sotasignat, ***Ramon Baldrich I Caselles***,
professor de l'Escola d'Enginyeria de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria ha
estat realitzat sota la seva direcció per

Alberto Valero Moreno

I per a que consti firma la present.
Sabadell, ***Setembre*** de ***2012***

Signat: ***Ramon Baldrich I Caselles***

FULL DE RESUM – PROJECTE FI DE CARRERA DE L'ESCOLA D'ENGINYERIA

Títol del projecte: DETECCIÓ DE MATRÍCULES AMB SISTEMES DE BAIX COST	
Autor: Alberto Valero Moreno	Data: Setembre 2012
Tutor: Ramon Baldrich i Caselles	
Titulació: Enginyeria Tècnica en Informàtica de Sistemes	
Paraules clau <ul style="list-style-type: none">• Català: APNR, OCR, reconeixement de caràcters, matrícules, Matlab• Castellà: APNR, OCR, reconocimiento de caracteres, matrículas, Matlab• Anglès: APNR, OCR, character recognition, license plate, Matlab	
Resum del projecte <ul style="list-style-type: none">• Català: En aquest projecte s'avaluen els mètodes utilitzats per els mecanismes de detecció de matrícules, i es proposarà un algorisme de detecció de matrícules dissenyat específicament per a sistemes de baix cost com ara els ordinadors actuals. Utilitzarem eines al nostre abast, com càmeres fotogràfiques domèstiques o de mòbil per analitzar el rendiment de l'algorisme.• Castellà: En este proyecto se evalúan los métodos utilizados por los mecanismos de detección de matrículas, y se propondrá un algoritmo de detección de matrículas diseñado específicamente para sistemas de bajo coste como los ordenadores actuales. Usaremos herramientas a nuestro alcance, como cámaras fotográficas domésticas o de móvil para analizar el rendimiento del algoritmo.• Anglès: This project evaluates the methods used by the license plate detection mechanisms, and proposes a license plate detection algorithm designed specifically for low cost systems such as today computers. We will use common tools such as domestic or mobile phone cameras to analyze the performance of the algorithm.	

Índex

1. Introducció	1
1.1. Presentació	1
1.2. Motivacions	2
1.3. Objectius	3
1.4. Estat de l'art.....	3
1.5. Estructura de la memòria	4
 2. Estudi de viabilitat	 5
2.1. Introducció	5
2.2. Anàlisi de requisits	7
2.3. Parts Interessades	9
2.4. Alternatives i selecció de la solució	10
2.5. Planificació del projecte	13
2.6. Riscos	15
2.7. Pressupost	17
2.8. Viabilitat Legal	18
2.9. Conclusions	19
 3. Disseny	 20
3.1. Especificacions de les matrícules	20
3.2. Fonaments i disseny del reconeixement de matrícules ...	22
3.3. Dissenys previs	33
3.4. Selecció del llenguatge	34
 4. Implementació	 37
4.1. Adquisició de la imatge i processament previ	37
4.2. Extracció de la regió de la matricula	39
4.3. Reconeixement individual de caràcters	46
4.4. Validació de la resposta	50
4.5. Tractament de les dades	50

5. Resultats	51
5.1. Precisió de l'algorisme	51
5.2. Temps d'execució de l'algorisme	55
5.3. Conclusió dels resultats	58
 6. Conclusions i treballs futurs	 60
6.1. Desviació de la planificació	60
6.2. Conclusions	61
6.3. Ampliacions	62
 7. Bibliografia	 63
7.1. Publicacions consultats	63
7.2. Referències electròniques consultades	64

Índex de taules

Taula 1: Definicions, abreviatures i acrònims	5
Taula 2: Priorització dels objectius	6
Taula 3: Priorització de requisits	8
Taula 4: Relació entre objectius i requisits	8
Taula 5: Stakeholders	9
Taula 6: Perfils d'usuari	9
Taula 7: Equip del projecte	9
Taula 8: Descripció d'alternatives	12
Taula 9: Recursos humans	13
Taula 10: Tasques del projecte	14
Taula 11: Catalogació dels riscos	16
Taula 12: Plans de contingència	16
Taula 13: Desglossament de costos	17
Taula 14: Estimació de cost del personal	17
Taula 15: Característiques de les matrícules espanyoles	21
Taula 16: Matriu de confusió de l'algorisme	52
Taula 17: Matriu de confusió en entorns de baixa il·luminació	52
Taula 18: Distribució d'errors de la detecció de la matrícula	53
Taula 19: Temps d'execució en M1	55
Taula 20: Temps d'execució en M2	56

Índex de figures

Fig. 1: Planificació temporal prevista	14
Fig. 2: Esquema del procés de reconeixement de matrícules	22
Fig. 3: Esquema de la extracció de la regió de la matrícula	24
Fig. 4: Imatge en escala de grisos	25
Fig. 5: Imatge millorada	25
Fig. 6: Imatge Bottom-Hat	26
Fig. 7: Imatge binaritzada	26
Fig. 8: Imatge amb els Bounding Box	27
Fig. 9: Aplicació de la transformada de Hough	28
Fig. 10: Contorns de les caixes	28
Fig. 11: Caixes amb les línies obtingudes	28
Fig. 12: Matrícula amb rectes de Hough	29
Fig. 13: Matrícula orientada	29
Fig. 14: Regió de la matricula	29
Fig. 15: Matricula detectada	30
Fig. 16: Esquema del reconeixement individual de caràcters	31
Fig. 17: Plantilla	31
Fig. 18: Lletra de la matricula	31
Fig. 19: Esquelet original i detectat	32
Fig. 20: Punts extrems original i detectat	32

Fig. 21: Punts de creuament original i detectat	32
Fig. 22: Imatge original	37
Fig. 23: Imatge redimensionada	38
Fig. 24: Bottom-Hat	39
Fig. 25: Binarització	40
Fig. 26: Imatge amb Bounding Box	41
Fig. 27: Imatge sense àrees petites i grans	41
Fig. 28: Imatge amb els contorns de les caixes	42
Fig. 29: Representació polar de la matriu acumuladora	43
Fig. 30: Rectes de Hough amb els Bounding Box	43
Fig. 31: Caixes seleccionades	44
Fig. 32: Regió on es troba la matrícula	45
Fig. 33: Caràcters de la matrícula	45
Fig. 34: Matrícula final detectada	46
Fig. 35: Objectes redimensionats amb el caràcter reconegut	47
Fig. 36: Diagrama de consens	49
Fig. 37: Errors entre les lletres originals i lletres reconegudes	54
Fig. 38: Temps d'execució en M1	55
Fig. 39: Temps d'execució en M2	56
Fig. 40: Gràfica de temps segons resolució	57
Fig. 41: Gràfica de coeficient de temps per píxel	58
Fig. 42: Planificació final i planificació prevista	60

1. Introducció

1.1. Presentació

Des de sempre, s'ha tingut la necessitat d'accelerar els processos burocràtics per tal de reduir el temps de resolució i tractament de les tramitacions. Amb el pas dels anys, els ordinadors han millorat sensiblement, permetent que tasques abans gairebé impensables es puguin resoldre ràpidament i amb una eficàcia difícil d'igualar. Amb això, diversos camps de la informàtica han progressat de tal manera que s'han pogut aplicar els nous avenços tecnològics per facilitar aquestes tasques burocràtiques pesades i difícils, com ara bé classificar dades personals en bases de dades, crear xarxes segures per compartir informació entre administracions, etc.

Una de les àrees amb més interès en fer servir aquesta capacitat és en el camp de la Visió per Computador, programar un ordinador per analitzar documents o escenes i tractar les dades extretes. Les aplicacions d'aquest camp van des de aplicacions mèdiques per fer diagnòstics, processos industrials on inspeccionen els productes finals per trobar defectes o detectar la orientació per controlar un braç robòtic, fins a aplicacions militars de reconeixement d'escenes per extreure dades estratègiques.

Una altra aplicació molt comú de la visió per computador és el Reconeixement Automàtic de Matrícules. S'utilitza una càmera per reconèixer matrícules de cotxes en tot tipus de situacions, com gestionar l'accés a pàrkings i peatges, o ara bé controlar la velocitat dels vehicles per una via. El reconeixement de matrícules ha evolucionat considerablement gràcies a elements com càmeres o sistemes de il·luminació especials, permetent nivells de identificació mai vistos, resultant gairebé infalibles.

Aquests sistemes no són econòmics, ni estan a l'abast de la majoria. Per tant, es poden buscar mètodes alternatius per tal de trobar un equilibri entre eficiència econòmica i eficàcia per obtenir la major rendibilitat al sistema.

Aquest projecte esta orientat a implementar un prototipus de sistema simple de reconeixement de matrícules fent servir equipament quotidià, com càmeres de mòbil i alhora adaptat per ser executat en ordinadors de baix cost actuals.

1.2. Motivacions

Hi ha hagut varies motivacions a l'hora d'escollir desenvolupar aquest projecte.

La primera és que sempre m'ha interessat el camp de la visió per computador i la intel·ligència artificial. Fer servir un ordinador per tal de que siguin capaços d'aprendre i fer operacions que normalment només serien capaços de realitzar persones sempre m'ha fascinat. A més a més, cada cop és més evident que el futur de la informàtica i de els aparells electrònics introdueixen elements d'aquests dos camps per construir eines abans impensables, i per tant el futur de la informàtica integrarà cada cop més intel·ligència artificial i Visio per Computador per a les innovacions tecnològiques.

Una altra motivació, lligada amb la primera, en l'assignatura de Visio Artificial varem realitzar una pràctica sobre detecció de la regió d'una matricula en imatges predefinides. Per aquest motiu em va semblar una bona idea poder ampliar els coneixements i aprofundir encara més en aquest tema, creant un projecte de reconeixement complet (des de el processament de la imatge fins al reconeixement dels caràcters) de matrícules.

A més a més, aprendre a construir un projecte des de zero, conèixer entorns de desenvolupament complets com Matlab i les seves eines, i els procediments de visió artificial suposen un primer contacte amb els projectes d'empreses del futur.

Per tant, l'aprenentatge de les eines, desenvolupar el projecte en aquestes àrees de la computació i ampliar el treball fet en l'assignatura em semblava un gran repte, i per això vaig decidir treballar en aquesta línia de projecte.

1.3. Objectius

A partir dels fets exposats en aquesta introducció i donada la importància de la detecció automàtica de matrícules, ens proposem, com a objectiu general, implementar un prototipus de sistema de reconeixement de matrícules, des de que la càmera pren la imatge fins que el programa la reconeix i la desa.

Tanmateix, també serà essencial que sigui un algorisme general i ràpid per tal de poder-se utilitzar en màquines de baix recursos actuals i càmeres fotogràfiques com les que tenim avui dia.

1.4. Estat de l'art

Els sistemes de reconeixement de matrícules estan a l'ordre del dia, tant per detectar matrícules de cotxes en moviment com per controlar la entrada de cotxes en zones tancades. Aquests sistemes tenen un alt grau d'èxit en la seva tasca, però són sistemes costosos econòmicament i molt cars de mantenir. Utilitzen sistemes hardware específics on inclouen sistemes d'il·luminació infraroja, sensors i càmeres especials per localitzar la posició del cotxe i sistemes de processament de dades dissenyats exclusivament per la tasca de detectar i processar matrícules.

Altres mètodes de detecció de matrícules es basen en dissenyar programes complets de detecció de matrícules per una entrada de vídeo o fotografies de la matricula. Aquests programes detecten una gran quantitat de plaques, però són bastant restrictius en la il·luminació i en l'escena, com ara que la matricula es trobi a una distància concreta, que no aparegui res més a l'escena que no sigui el cotxe, etc. Tot això impedeix que es pugui aplicar en zones diferents o espais oberts amb gran quantitat d'elements a l'escena i no és possible millorar aquestes restriccions ja que són codis tancats.

Com es veurà més endavant, aquestes solucions no s'ajusten als objectius marcats del projecte, i per tant s'investigaran altres alternatives a aquestes per tal d'assolir les fites.

1.5. Estructura de la memòria

La resta de la memòria estarà estructurada en els següents apartats:

Estudi de viabilitat

Analitzarem el projecte i identificarem els requisits de la aplicació i les restriccions del sistema. Tot seguit estudiarem totes les alternatives que tenim, en seleccionarem una i avaluarem la viabilitat del projecte.

Disseny

En aquest capítol exposarem els coneixements previs necessaris per la comprensió del projecte i el disseny de l'algorisme. Mostrarem els dissenys anteriors de l'algorisme i tot seguit analitzarem els llenguatges més adients per la implementació.

Implementació

Explicarem com s'han implementat els diferents mòduls de la aplicació pas per pas fins al resultat final.

Resultats

Presentarem i analitzaran els resultats obtinguts del reconeixement de matrícules, mesurant tant la precisió com el temps d'execució.

Conclusions i treballs futurs

Presentarem les conclusions a les que hem arribat al finalitzar el projecte i obrirem diverses línees de treballs futurs que ampliaran la feina ja presentada.

Bibliografia

Aquest apartat conté totes les referències bibliogràfiques consultades per la realització del projecte, tant electròniques com físiques.

2. Estudi de Viabilitat

En aquesta secció es farà un anàlisi exhaustiu del projecte, tant a nivell tècnic dels requisits de la aplicació com les alternatives per complir aquests requisits. Finalment escollirem una de les alternatives, i analitzarem la viabilitat econòmica i la seva planificació del treball.

2.1. Introducció

Tipologia: desenvolupament.

Paraules clau: APNR, OCR, reconeixement de caràcters, matrícules.

2.1.1. Descripció

Degut a que els sistemes de reconeixement de matrícules son cars, difícils de mantenir o d'adaptar a entorns oberts i precisen de eines especialment adaptades a aquesta tasca, es vol crear un prototipus d'algorisme per reconèixer matrícules en entorns diversos, utilitzant eines al nostre abast com càmeres digitals domèstiques o càmeres web.

2.1.2. Definicions, abreviatures i acrònims

En aquest quadre presentarem les abreviatures, definicions i acrònims que utilitzarem al llarg de la memòria del projecte.

ANPR	Automatic Number Plate Recognition, sistema automàtic de reconeixement de matrícules
Bounding Box	Caixa contenidora, són els punts que delimiten un objecte formant un rectangle amb la mínima dimensió
Closing	Operació morfològica que ens junta i tanca objectes molt propers segons l'element estructurant seleccionat.
Element estructurant	Matriu que utilitzem en diverses operacions morfològiques.

LOPD	Llei Orgànica de Protecció de Dades
Matlab	Entorn de desenvolupament d'aplicacions.
MP	Megapíxels, mesura del total de píxels d'una imatge concreta. 8 megapíxels equivalen a 8 milions de píxels.
OCR	Optical Character Recognition, reconeixement òptic de caràcters
SDK	Software Development Kit, un conjunt d'eines per desenvolupar software amb uns paquets i metodologia predefinita

Taula 1: Definicions, abreviatures i acrònims.

2.1.3. Objectius del projecte

Els principals objectius del projecte són:

O1: Reconeixement de matrícules estàtiques amb un grau d'èxit acceptable.

O2: Implementació adient per a sistemes de baix cost.

O3: Executable en sistemes operatius diferents.

O4: Capacitat de reconeixement sota diferents il·luminacions.

O5: Capacitat de reconèixer matrícules en moviment.

O6: Capaç de captar les imatges per una càmera instal·lada.

En la següent taula es pot veure la prioritització dels objectius:

	O1	O2	O3	O4	O5	O6
Principal	X	X		X		X
Secundari			X			
Opcional					X	

Taula 2: Priorització dels objectius.

2.2. Anàlisi de requisits

En aquesta secció valorarem els requisits del nostre projecte, tant funcionals com no funcionals, i les restriccions que s'imposen en el sistema.

2.2.1. Requisits funcionals

Els requisits funcionals són les especificacions del funcionament intern de la aplicació. Els requisits són:

RF1: Reconeixement de matrícules estàtiques (cotxes estacionats o parats).

RF2: Reconeixement de matrícules dinàmiques (cotxes en moviment).

RF3: Capacitat de guardar la matricula per poder accedir posteriorment.

RF4: Capacitat d'executar-se en màquines de baix cost actuals.

2.2.2. Requisits no funcionals

Els requisits no funcionals són aquells que influeixen en el disseny de la aplicació o en la implementació d'aquesta. Els requisits no funcionals que componen aquest projecte són els següents:

RNF1: Velocitat de reconeixement acceptable.

RNF2: Capacitat d'execució en màquines amb baixos recursos (contemplant les restriccions del sistema).

RNF3: Multiplataforma, capaç d'executar-se en diversos sistemes operatius.

RNF4: Compliment de la LOPD en referència a guardar les dades.

2.2.3. Restriccions

Són especificacions del projecte que limiten l'entorn de treball, on el propi programa presenta unes restriccions en l'escena i la il·luminació, que són les següents:

- Fotografia de la matrícula entre 2 i 5 metres.
- Il·luminació suficient, on els caràcters de la matrícula quedin contrastats amb el fons de la placa. Per fotografies nocturnes, sistema d'il·luminació que permeti la correcta visualització de la matrícula.
- Una única matrícula per fotografia.
- Matrícules espanyoles homologades dels següents tipus ordinària llarga, ordinària llarga davantera, ordinària llarga amb sigles de província, ordinària llarga davantera amb sigles de província (veure taula 14).

2.3.4. Priorització de requisits

	RF1	RF2	RF3	RF4	RNF1	RNF2	RNF3	RNF4
Crític	X		X	X	X	X		X
Condicional								
Opcional		X					X	

Taula 3: Priorització de requisits.

2.3.5. Relació entre objectius i requisits

	RF1	RF2	RF3	RF4	RNF1	RNF2	RNF3	RNF4
O1	X		X	X	X	X		X
O2	X	X	X	X	X	X		
O3			X	X			X	
O4	X	X	X	X	X	X		X
O5		X	X	X				X
O6	X	X	X	X		X		X

Taula 4: Relació entre objectius i requisits.

2.4. Parts interessades

A continuació exposarem les parts interessades del projecte, així com l'equip responsable del seu desenvolupament i les tasques que han realitzat.

2.4.1. Stakeholders

Nom	Descripció	Responsabilitat
Ramon Baldrich i Caselles	Director del projecte	Supervisa el treball de l'alumne.
Alberto Valero Moreno	Responsable del projecte	Realitza el projecte amb la supervisió del director del projecte.

Taula 5: Stakeholders.

2.4.2. Perfils d'usuari

Nom	Descripció	Responsabilitat
Administrador	Administrador del sistema	Administra i manté el sistema
Usuari	Usuari comú	Utilitza el sistema.

Taula 6: Perfils d'usuari.

2.4.3. Equip del projecte

Càrrec	Descripció	Responsabilitat
Tutor del projecte	Ramon Baldrich i Caselles	Supervisa el treball de l'alumne.
Cap del projecte	Alberto Valero Moreno	Defineix, gestiona, planifica i controla el projecte.
Analista	Alberto Valero Moreno	Col·labora amb el cap del projecte. Realitza el disseny, especificacions i requisits.
Programador	Alberto Valero Moreno	Desenvolupa l'algorisme segons l'anàlisi previ i els requisits. Participa en les proves del programa.

Tècnic de proves	Alberto Valero Moreno	Realitza les proves adients per corregir els errors del programa.
------------------	-----------------------	--

Taula 7: Equip del projecte.

2.5. Alternatives i selecció de la solució

En aquesta secció s'estudiaran les diverses alternatives que compleixin els requisits del projecte i escollirem la millor.

2.5.1. A1: Alternativa 1

Instal·lar i adaptar un sistema complet de reconeixement de matrícules.

Aquesta opció es basa en comprar un sistema ja preparat i optimitzat per l'ús que volem fer, juntament amb el seu programari. S'haurà d'instal·lar en l'entorn de treball segons les especificacions del fabricant, i el funcionament es redueix a les funcionalitats que dona el sistema.

Funcionalitats:

- Reconeixement de matrícules amb un grau alt d'èxit.
- Reconeixement assistit amb aprenentatge.
- Sistema avançat de captura d'imatges, amb càmeres especials.
- Sistema PC adaptat per al treball de dades.

Deficiències:

- Adaptable només segons les especificacions del fabricant, normalment restrictives.
- Necessitat d'instal·lació i manteniment regular.
- Necessitat d'un equip especial costós.

2.5.2. A2: Alternativa 2

Comprar un Software Development Kit (SDK) de reconeixement de matrícules i programar les funcions, com per exemple CARMEN(r) FreeFlow [9].

Aquesta opció utilitzaria un conjunt de funcions ja programades específicament per la tasca de reconèixer matrícules en diversos entorns, programades en un llenguatge específic. No és necessari cap tipus de equip especial de captura d'imatges, però sí necessita una targeta de vídeo per la entrada de la imatge.

Funcionalitats:

- Reconeixement de matrícules amb un grau mitjà/alt d'èxit.
- Reconeixement en entorns estàtics o dinàmics.
- Independent de l'equip de captura.

Deficiències:

- No és possible millorar el temps d'execució en les funcions específiques del SDK, ja que són funcions protegides.
- Funcions acotades a les especificacions del SDK, possibilitat d'implementació pròpia.
- Problema amb l'ús comercial o privat segons les llicències i exigències dels creadors.
- Temps de reconeixement de matrícula relativament alts per ordinadors de prestacions mitjanes.

2.5.3. A3: Alternativa 3

Creació i implementació d'un sistema de reconeixement de matrícules

La opció proposada és la creació d'un programa complet de reconeixement de caràcters. És una solució adaptable i modificable, en un llenguatge ja conegut i fàcil d'aprendre. No faria falta cap tipus d'equip especial de captura d'imatges.

Funcionalitats:

- Reconeixement de matrícules amb un grau mitjà/alt d'èxit.
- Adaptable a sistemes de baix cost, reduint el consum de recursos del sistema.
- Adaptable a entorns específics de captura d'imatges.
- Independent de l'equip de captura.

Deficiències:

- Sistema no tan robust com altres alternatives.

2.5.4. Solució proposada

En la següent taula estan descrites les alternatives i les seves característiques

Alternativa	Cost de l'equip	Cost d'instal·lació	Cost de desenvolupament
A1	Molt Alt	Alt	Mitjà
A2	Baix	Baix	Alt
A3	Mínim	Baix	Alt
Alternativa	Cost de manteniment	Grau d'èxit	Rendiment
A1	Alt	Alt	Alt
A2	Baix	Mitjà	Baix
A3	Baix	Mitjà	Alt

Taula 8: Descripció d'alternatives.

Ja que un dels objectius crítics del projecte és un bon rendiment en màquines amb recursos limitats, i també és necessari que sigui adaptable a l'equip amb baix cost, limitem les alternatives a la A3.

2.6. Planificació del projecte

El projecte es començarà a desenvolupar al febrer del 2011 i es finalitzarà al juny del 2011 amb una dedicació de 4 hores diàries. El total d'hores dedicades al projecte serà de 300 hores aproximadament

Data de començament: 1 de Febrer de 2011

Data de finalització: 9 de Juny de 2011

Eines de planificació i control: Microsoft Project 2003

2.6.1. Recursos del projecte

Per fer una estimació del pressupost és necessari indicar el cost per hora de cada recurs que utilitzem. Això ho podrem observar en la següent taula:

Recurs humà	Valoració
Cap de projecte	50 €/h
Analista	30 €/h
Programador	18 €/h
Tècnic de proves	12 €/h

Taula 9: Recursos humans.

2.6.2. Tasques del projecte

A continuació es defineixen les tasques que s'han de realitzar dintre el període per desenvolupar l'algorisme, amb la durada aproximada de cada tasca i especificant els recursos utilitzats:

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	[-] Detecció de matrícules en sistemes de baix cost	316 horas	mar 01/02/11	jue 09/06/11		
2	[-] Planificació	24 horas	mar 01/02/11	mié 09/02/11		
3	Anàlisi de requisits	16 horas	mar 01/02/11	vie 04/02/11		Cap de projecte
4	Cerca d'alternatives	8 horas	lun 07/02/11	mié 09/02/11	3	Analista
5	[-] Disseny	80 horas	mié 09/02/11	mar 15/03/11	2	
6	[-] Estudi dels fonaments teòrics	40 horas	mié 09/02/11	vie 25/02/11	4	
7	Estudi de l'estat de l'art	20 horas	mié 09/02/11	jue 17/02/11		Analista
8	Estudi de processos de visió artificial	20 horas	jue 17/02/11	vie 25/02/11	7	Analista
9	Disseny de la solució	36 horas	vie 25/02/11	vie 11/03/11	6	Analista
10	Selecció de llenguatge i recursos	4 horas	vie 11/03/11	mar 15/03/11	9	Analista
11	[-] Implementació	120 horas	mar 15/03/11	mar 03/05/11	9	
12	Implementar la localització de matrícula	60 horas	mar 15/03/11	jue 07/04/11	10	Programador;Analista[33%]
13	Implementar la lectura de caràcters	60 horas	jue 07/04/11	mar 03/05/11	12	Programador;Analista[33%]
14	[-] Proves	54 horas	mar 03/05/11	mié 25/05/11	5	
15	Recollir matrícules per a les proves	12 horas	mar 03/05/11	vie 06/05/11	11	Tècnic de proves
16	Correcció d'errors	32 horas	vie 06/05/11	vie 20/05/11	15	Tècnic de proves;Programador
17	Anàlisi de resultats i rendiment	10 horas	vie 20/05/11	mié 25/05/11	16	Tècnic de proves;Analista
18	Realització de la memòria	34 horas	mié 25/05/11	mié 08/06/11	14	Cap de projecte
19	Tancament del projecte	4 horas	mié 08/06/11	jue 09/06/11	18	Cap de projecte

Taula 10: Tasques del projecte.

2.6.3. Tasques del projecte

En la següent figura podem observar la planificació prevista del projecte, segons les tasques de la taula anterior amb la seva durada aproximada:

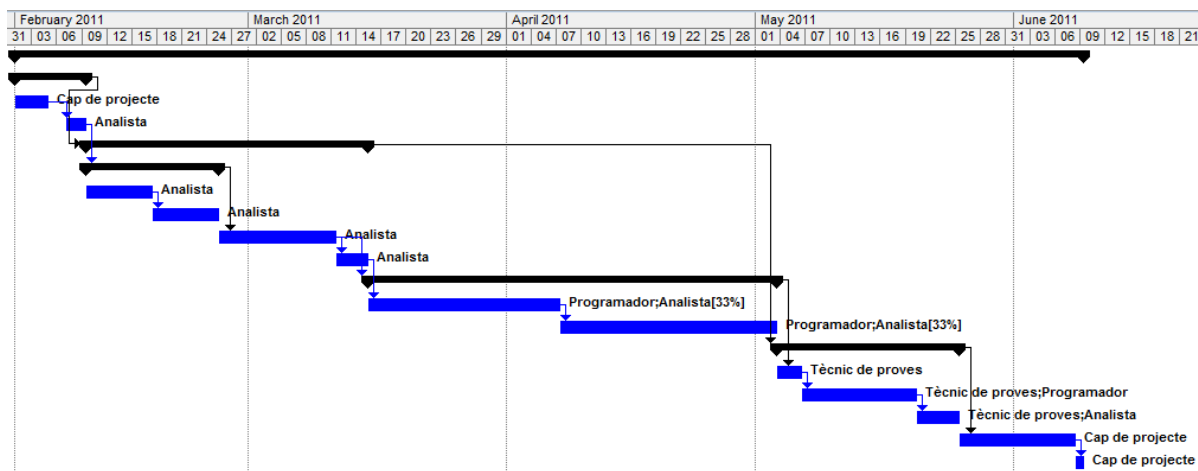


Fig. 1: Planificació temporal prevista.

2.7. Riscos

Els riscos desenvolupant el programa poden afectar a tot el projecte sencer i fer diferir els anàlisis i planificacions del resultat final.

2.7.1. Avaluació de riscos

R1. Tecnologia triada inadequada: Que no s'adapti als requisits de l'aplicació i no es puguin complir els objectius.

R2. Planificació temporal optimista: Ens ho provocaria un estudi de viabilitat optimista que allargaria el procés de desenvolupament del projecte.

R3. Canvi en els requisits: Ens ho provocaria si haguéssim de fer canvia en l'apartat de requisits de l'estudi de viabilitat, que enrederaria la data de finalització del projecte.

R4. Pressupost poc ajustat: Si el pressupost esta mal ajustat, ens podria provocar pèrdues econòmiques o que la qualitat fos inferior.

R5. Fase de test incorrecta: Pot aparèixer en la fase de desenvolupament i implementació. Pot provocar una pèrdua de qualitat, insatisfacció dels objectius i requisits i en definitiva, afectar a tot el programa.

R6. Incompliment de les normes: Pot aparèixer en qualsevol fase del projecte degut que no es complissin les normes establertes segons la LOPD i pot tenir repercussions legals.

R7. Abandonament del projecte: Pot aparèixer en qualsevol fase. Pot provocar pèrdues econòmiques importants.

2.7.2. Catalogació dels riscos

Risc	Probabilitat	Impacte
R1	Baixa	Crític
R2	Alt	Crític
R3	Alt	Marginal
R4	Alt	Crític
R5	Baix	Crític
R6	Alt	Crític
R7	Mitjà	Catastròfic

Taula 11: Catalogació dels riscos.

2.7.3. Plans de contingència

Si aparegués algun risc mencionat prèviament les possibles mesures de contingència serien les següents:

Risc	Solució que cal adoptar
R1	Buscar tecnologies alternatives que s'ajustin als requisits i objectius. Tornar a programar les parts que ho necessitin.
R2	Enrederir alguna funcionalitat. Afrontar les possibles pèrdues.
R3	Enrederir o modificar la planificació i el pressupost.
R4	Afrontar les pèrdues ocasionades i minimitzar-les al màxim.
R5	Ajustar la planificació de proves i fer la major quantitat de proves possible.
R6	Aplicar polítiques de seguretat per restringir l'accés de les dades.
R7	No té solució.

Taula 12: Plans de contingència.

2.8. Pressupost

En aquesta secció farem un càlcul aproximat dels costos que impliquen el personal i els recursos necessaris per la realització del projecte.

2.8.1. Estimació de costos dels recursos

Realitzem el nostre projecte en un entorn Windows 7 i el software de desenvolupament serà Matlab 2010a. Les suites de Matlab tenen un preu variable segons l'aplicació i els paquets de la versió, però en el nostre cas, no fem el projecte amb orientació professional i la versió d'estudiant és suficient, afegint el paquet de Image Acquisition Toolbox per prendre imatges de les càmeres. Podem veure el desglossament del preu de cada punt en la següent taula:

Producte	Unitats	Preu unitari	Cost
PC	1	600€	600€
Windows 7	1	119€	120€
MS Office	1	249€	250€
MS Project	1	360€	360€
Matlab Student	1	820€	820€
Matlab Image Acquisition Toolbox	1	800€	800€
Càmera	1	100€	100€
TOTAL			3050€

Taula 13: Desglossament de costos dels recursos.

2.8.1. Estimació de costos del personal

A continuació exposem el cost total per cada rol i les hores dedicades:

Recurs Humà	Hores de treball	Cost total
Cap de projecte	54 hores	2700 €
Analista	138 hores	4140€
Programador	152 hores	2432€
Tècnic de proves	54 hores	648€
TOTAL		9920 €

Taula 14: Estimació de cost del personal.

2.9. Viabilitat legal

Per considerar aquest punt, hem d'estudiar la Llei Orgànica de Protecció de Dades per tal de veure si les matrícules de cotxe suposen una dada personal, i per tant s'han de seguir uns processos per mantenir la seguretat.

2.9.1 Llei Orgànica de Protecció de Dades

Segons la Agencia Espanyola de Protecció de Dades (AEPD), el tractament de les matrícules de cotxe està inclòs dins del marc de dades personals, ja que la identificació de la persona es pot fer sense fer grans esforços ni temps desproporcionats [6]. Per tant, el nostre projecte haurà de complir la Llei Orgànica de Protecció de Dades (LOPD) [7].

La LOPD és una llei que té com a funció garantir i protegir les llibertats públiques i els drets fonamentals de les persones físiques, en particular el dret a l'honor i a la intimitat personal i familiar. Només és aplicable a les persones físiques.

La LOPD estableix 3 nivells de seguretat segons les dades que es guardin:

- Nivell bàsic: qualsevol fitxer que contingui dades personals.
- Nivell mig: qualsevol fitxer que contingui dades relatives a infraccions administratives o penals, serveis financers i solvència patrimonial i crèdit.
- Nivell alt: qualsevol fitxer que a més contingui dades sobre ideologia, religió, creences, origen racial, salut, vida sexual i dades amb finalitats policials sense el consentiment de les persones afectades.

El nostre projecte està classificat en el nivell bàsic. Això comporta que es segueixin una sèrie de normes per adaptar-se a la llei:

- Inscripció dels fitxers de dades a la Agencia Espanyola de Protecció de Dades.
- Creació d'un document de seguretat (es pot descarregar de la web de la AEPD).
- Sistema de control d'accés als fitxers que continguin les matrícules.

En cas que no es complissin aquestes normes, es podria incórrer en alguna falta, amb multes que van des de els 600 als 600.000 € depenent de la gravetat de la falta.

2.10. Conclusions

Tenint en compte tota la informació exposada en aquest apartat podem extreure les conclusions de l'estudi de viabilitat.

En primer lloc s'ha realitzat un estudi dels objectius per cobrir les nostres necessitats, i les alternatives que podem trobar avui dia al mercat són escasses, i no ens proporcionen els nostres requeriments de manera completa, o bé ens contradiu els objectius que hem marcat.

D'altra banda, s'ha realitzat un estudi dels costos i riscos i hem determinat que donat als objectius que volem són assumibles ja que per la tipologia de projecte són raonables i el cost més elevat és el cost de persona/hora.

Finalment, podem concloure que el projecte és viable i es procedeix a la seva realització.

3. Disseny

En aquest capítol tractarem els fonaments teòrics bàsics per entendre el projecte així com el disseny de la aplicació i els dissenys previs fins a la solució final. També farem un estudi i selecció del llenguatge en que implementarem el programa.

3.1. Especificacions de les matrícules

Abans de comentar els aspectes teòrics del reconeixement de matrícules, hem de definir com són les matrícules espanyoles que detectarem. Si es vol ampliar la informació referiu-vos a [8].

Ara per ara, coexisteixen dos sistemes de matriculació. El primer, introduït el 1971, es compon de 1 o 2 lletres inicials identificadores de la província de matriculació, 4 xifres numèriques, i 2 lletres finals, fent un total de 7 ó 8 caràcters. Les dimensions són de 520 x 110 mm amb lletres de 45 x 77 mm.

El segon sistema, introduït l'any 2000, utilitza un sistema centralitzat de matriculació, compost per 4 xifres i 3 lletres finals, fent un màxim de 7 caràcters. Aquestes matrícules porten un identificador nacional (una lletra E blanca sobre fons blau amb la bandera de la Unió Europea). Les dimensions són idèntiques a les anteriors, amb diverses variacions que es poden veure a la taula 12.

Totes les matrícules comparteixen la característica que tenen un fons blanc reflectant amb lletres negres mat. Aquesta característica ajuda a reconèixer les matrícules en entorns de baixa il·luminació.

Les matrícules que el nostre algorisme detectarà són:

- Ordinària llarga (520x110 mm).
- Ordinària llarga davantera (340x110 mm).
- Ordinària llarga amb sigles de província (520x110 mm).
- Ordinària llarga davantera amb sigles de província (340x110 mm).

Tipus de placa de matrícula	Mesures (mm)	Caràcters (mm)			Nombre de caràcters	Color	
		Mesures	Separació entre caràcters	Gruix del traç		Fons	Caràcters
Ordinària llarga	520 x 110	45 x 77	14	10	7	B	N
Ordinària alta	340 x 220	45 x 77	10	10	7	B	N
Ordinària llarga davantera	340 x 110	30 x 60	8	6	7	B	N
Motocicletes ordinària	220 x 160	30 x 60	10	6	7	B	N
Ordinària llarga amb sigles de província	520 x 110	45 x 77	8	10	7 ó 8	B	N
Ordinària alta amb sigles de província	340 x 220	45 x 77	10	10	7 ó 8	B	N
Ordinària llarga davantera amb sigles de província	340 x 110	30 x 60	6	6	7 ó 8	B	N
Motocicletes ordinària amb sigles de província	220 x 160	30 x 60	8	6	7 ó 8	B	N

Taula 15: Característiques de les matrícules espanyoles.

3.2. Fonaments i disseny del reconeixement de matrícules

El procés de reconeixement de matrícules es pot estructurar en 5 parts diferenciades:

1. Adquisició de la imatge i processament previ.
2. Extracció de la regió de la matrícula.
3. Reconeixement individual de caràcters.
4. Validació de la resposta.
5. Tractament de les dades extretes.

Es pot veure un esquema del procés a continuació:

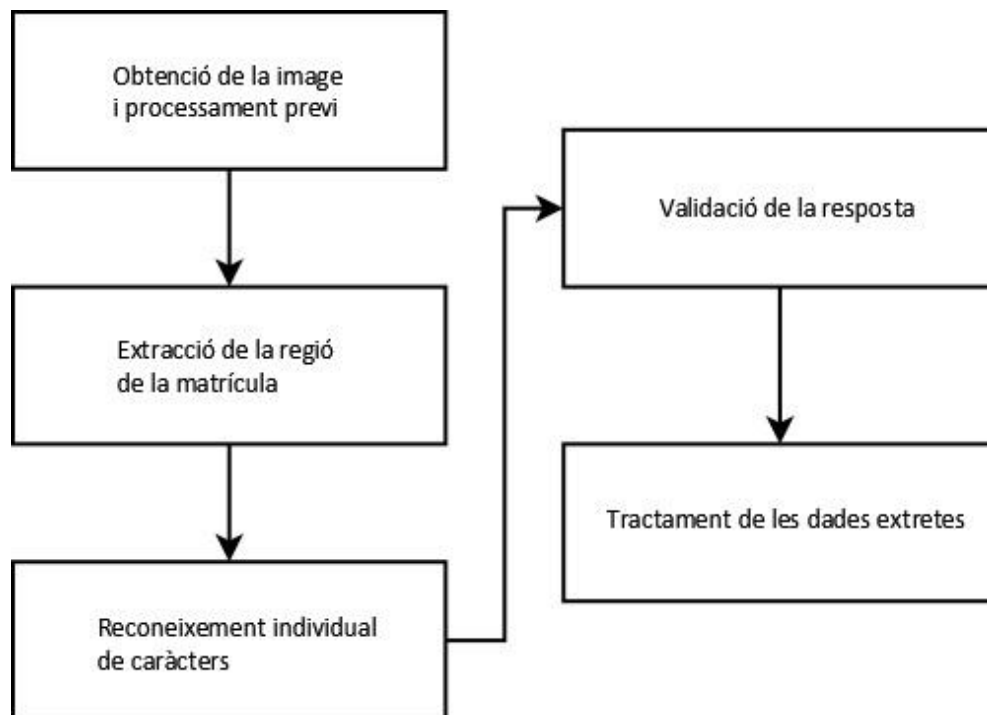


Fig. 2 Esquema del procés de reconeixement de matrícules

3.2.1. Adquisició de la imatge i processament previ

El procés de reconeixement de caràcters comença des de la adquisició de la imatge. Aquest procés es pot fer des de varies fonts:

- **Càmera digital:** És el tipus de càmera que hem pres en el projecte. Són càmeres molt econòmiques, la qualitat de la imatge és més que suficient per el procés de reconeixement i són fàcils d'obtenir. En podem distingir dos tipus: les càmeres fotogràfiques i les càmeres web. La primera simplement es dedica a obtenir les imatges, s'ha de desar en una carpeta i després l'algorisme reconeix les imatges. El segon tipus es pot connectar directament a l'ordinador on es reconeixen les imatges, i mitjançant un disparador (per exemple, pitjar un botó), el programa s'encarrega d'encendre la càmera, prendre la fotografia i detectar la matrícula automàticament.
- **Càmera de videovigilància:** Càmera adaptada a escenes concretes, amb diversos sistemes per poder detectar el moviment i visió nocturna mitjançant llum infraroja. Càmeres adients per ambients exteriors on és necessària la detecció tant de dia com de nit, però és molt costosa. Aquest és el tipus de càmera que s'utilitza en la tasca de reconeixement de matrícules en àmbits estatals. En el cas que es volgués utilitzar una, no farien falta gaires canvis en l'algorisme per fer-la funcionar, i molt possiblement els resultats que obtindríem en entorns de baixa il·luminació serien molt bons.

En qualsevol cas s'ha de tenir en compte al captar les imatges una sèrie de factors que cal evitar, com ara bé el desenfoc de la càmera, il·luminació pobre de l'escena o que un objecte s'interposi entre la matrícula i la càmera, com una barra del para-xocs, etc. La imatge ideal és aquella en que les lletres de la matrícula es destaquen clarament del fons de la matrícula, és perpendicular a la càmera i l'escena és la més simple possible per facilitar la feina a l'algorisme.

Amb tot i això, i segons els objectius que ens hem fixat en l'estudi de viabilitat, el nostre algorisme haurà de ser capaç de detectar matrícules amb diverses il·luminacions (imatges diürnes o nocturnes, amb diverses fonts de il·luminació o només il·luminació ambiental), així com amb diversos elements en la escena que no pertanyin ni a la matrícula ni al vehicle en sí. També haurà de ser capaç, en certa mesura, d'aconseguir reconèixer les matrícules que no estiguin perpendiculars a la càmera, o que no estiguin orientades al pla horitzontal.

En el nostre algorisme farem ús d'una càmera digital, i durem a terme processos previs per adaptar-ho als requisits de la aplicació. Primer obtindrem la imatge i la passem a gama de grisos, per tal d'agilitzar el processos següents i reduir el consum de memòria. Seguidament, redimensionarem la fotografia per tal de treballar amb una imatge molt més reduïda sense pèrdua d'informació.

3.2.2. Extracció de la regió de la matrícula

En aquest procés, ens fem servir de diferents mètodes per netejar i localitzar les imatges. Principalment utilitzarem operacions morfològiques sobre les imatges. El procés es compon d'aquests passos:

1. Transformació de la imatge a binària.
2. Trobar els objectes que semblen lletres.
3. Localitzar els objectes alineats i corregir la orientació.
4. Localitzar les zones que semblen matrícula.
5. Extracció de la matrícula.

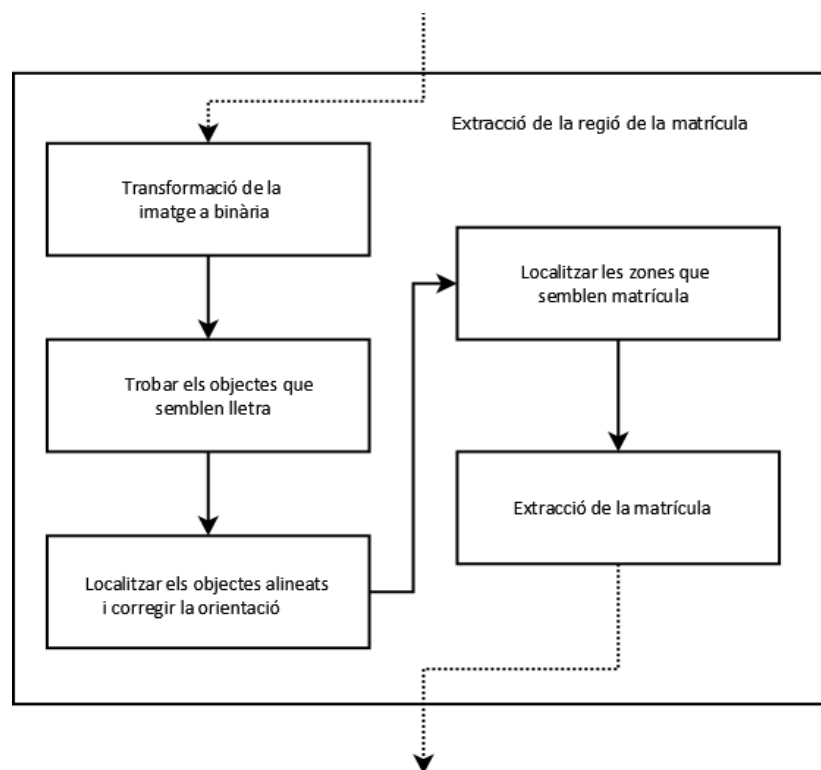


Fig. 3: Esquema de la extracció de la regió de la matrícula.

Transformació de la imatge a binària

Les imatges obtingudes poden estar fetes en una gran varietat d'entorns i il·luminacions diferents, on no tots ajuden a l'algorisme per detectar la matrícula. El primer procés que fem a la imatge serà augmentar el contrast entre blancs i negres, facilitant els passos següents. Es pot veure un exemple a continuació



Fig. 4: Imatge en escala de grisos.



Fig. 5: Imatge millorada.

Seguidament, farem servir un operador morfològic anomenat Bottom-Hat. Aquest procés utilitza una matriu que forma una màscara, anomenat element estructurant, que delimita una zona d'anàlisi que recorre la imatge per transformar-la. Essencialment, aquest procés extreu totes les zones fosques més petites que aquest element estructurant, com ara bé els caràcters de la matrícula, enfosquint la resta de zones. Per tant, ho utilitzarem per contrastar els caràcters de la matrícula respecte altres elements de la escena i de la pròpia placa.



Fig. 6: Imatge Bottom-Hat.

Seguit d'això, binaritzem la imatge. Escollirem un nivell baix, i els píxels que no arribin a aquest llindar, es quedaran a 0 (negre), la resta de píxels els marcarem a 1 (blanc). A partir d'aquest procés totes aquelles regions connectades amb píxels blancs els anomenarem objectes, mentre que la resta serà el fons.



Fig. 7: Imatge binaritzada.

Trobar els objectes que semblen lletres

Després del procés de binarització, seguim buscant objectes que semblin lletres. Fem això mirant la relació d'aspecte que tenen els objectes de la imatge. Si no compleixen les regles preestablertes (com hem dit abans, les lletres de les matrícules tenen el doble d'alçada que d'amplada), els eliminem de la escena.

L'operand utilitzat aquest cas, que ens serà molt útil en altres seccions, és la Bounding Box (caixa contenidora). Aquesta caixa té la propietat de ser el rectangle més petit que conté completament l'objecte. Amb les mesures de la caixa, ens és molt fàcil de mirar les proporcions de l'objecte, i de pintar aquesta caixa per a futures operacions.

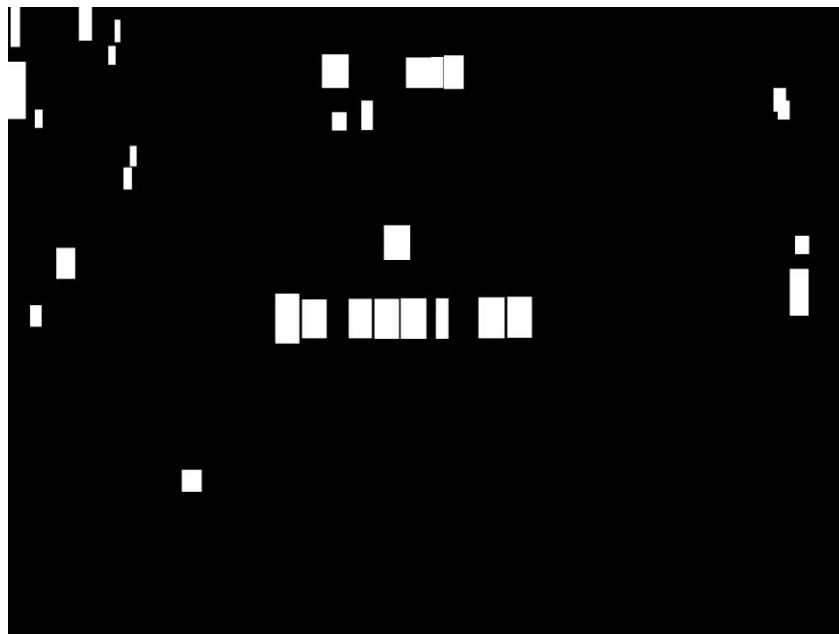


Fig. 8: Imatge amb els Bounding Box.

Localitzar els objectes alineats i corregir la orientació

En aquest procés ens servirem d'un algorisme molt utilitzat en la detecció de patrons: la transformada de Hough. El que volem obtenir és si existeix una o més rectes que uneixin aquests objectes, en el nostre cas els caràcters de la matrícula, i trobar tant la regió de la matrícula com la inclinació i profunditat de la matrícula per corregir-la.



Fig. 9: Aplicació de la transformada de Hough [1].

Per preparar la imatge trobarem els contorns de les caixes i passarem aquesta imatge a l'algorisme. Aquest s'encarregarà d'analitzar la imatge i trobarà totes les possibles rectes que uneixen tots els píxels de la imatge. Amb aquest resultat, mirarem quina ha sigut la recta més escollida i la desarem per analitzar les caixes que han contribuït en aquesta solució. Ja que no estem interessats en descobrir rectes verticals, limitem el rang d'inclinació d'aquestes rectes a 45 graus. També utilitzarem la propietat de les matrícules que les lletres es troben relativament juntes, per tant no considerarem els objectes massa allunyats

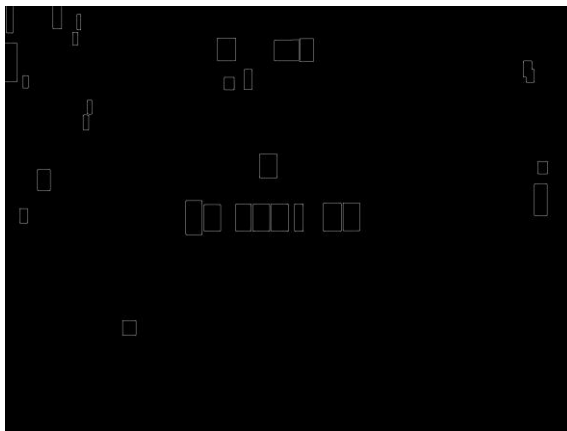


Fig. 10 Contorns de les caixes.

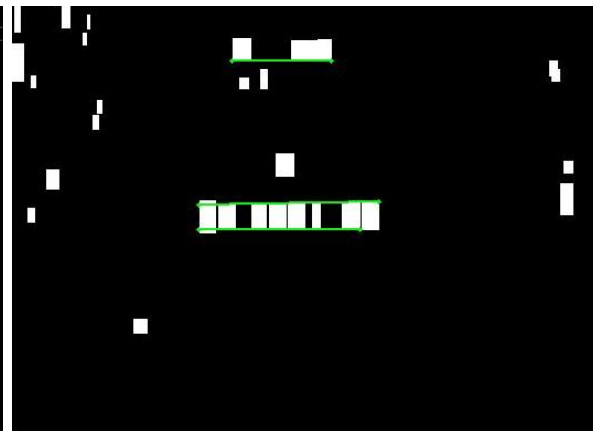


Fig. 11: Caixes amb les línies obtingudes.

Un cop trobades, utilitzarem aquestes línies per tal de detectar l'angle d'inclinació i la profunditat que té la imatge, i corregir-ho per tal de facilitar les operacions posteriors i millorar la detecció final de la matricula.



Fig. 12: Matricula amb rectes de Hough.



Fig. 13: Matricula orientada.

Finalment, de la imatge orientada detectarem aquells elements que no estan completament alineats, i per tant no són caràcters sinó artefactes de la detecció, com símbols o reflexos, i els eliminarem per als següents processos.

Localitzar les zones que semblen matricula

Seguidament, les lletres de les matrícules estan molt juntes, i els altres elements de la escena estan dispersos i inconnexos. Per tant, farem servir un procés, anomenat tancament o *closing*, que mitjançant un element estructurant ens unirà les lletres de la matricula que estan juntes i alineades, i els altres elements els mantindrà iguals. Farem servir un element estructurant suficient per connectar totes les lletres horitzontalment, i filtrarem els elements connectats d'aquesta manera per tal de complir les especificacions de matrícula.

Aquest cop, el filtratge ho farem segons una mesura del objectes, la excentricitat. La excentricitat és un paràmetre escalar que determina la forma de l'objecte, sent 0 si és una circumferència perfecta, o 1 si és un segment allargat. Amb això aconseguirem finalment la regió de la matrícula.



Fig. 14: Regió de la matricula.

Extracció de la matrícula

Un cop fet el filtratge, l'últim que ens queda és retallar la zona de la matrícula i iniciar el procés de reconeixement de caràcters.



Fig. 15: Matrícula detectada.

3.2.3. Reconeixement individual de caràcters

En aquest procés farem el reconeixement de les lletres i números de la matrícula. Aquest procés està molt lligat amb la validació de la resposta, per tal de comprovar els resultats i parar el reconeixement quan ja tenim totes les lletres reduir al màxim el temps d'execució. En la figura 16 es pot veure un diagrama del procés que realitza.

El mètode emprat per analitzar els caràcters de la matrícula és per correlació de plantilla. Una plantilla és una imatge prototipus, on el caràcter es veu de la millor forma possible (es pot veure una imatge d'una plantilla a la figura 17). Mitjançant aquesta plantilla, compararem un a un el caràcter extret amb aquestes plantilles, realitzant la suma del píxels que coincideixen. Les plantilles, per tal d'agilitzar el procés, són raonablement petites, i el cost computacional de comparar les plantilles és baix.

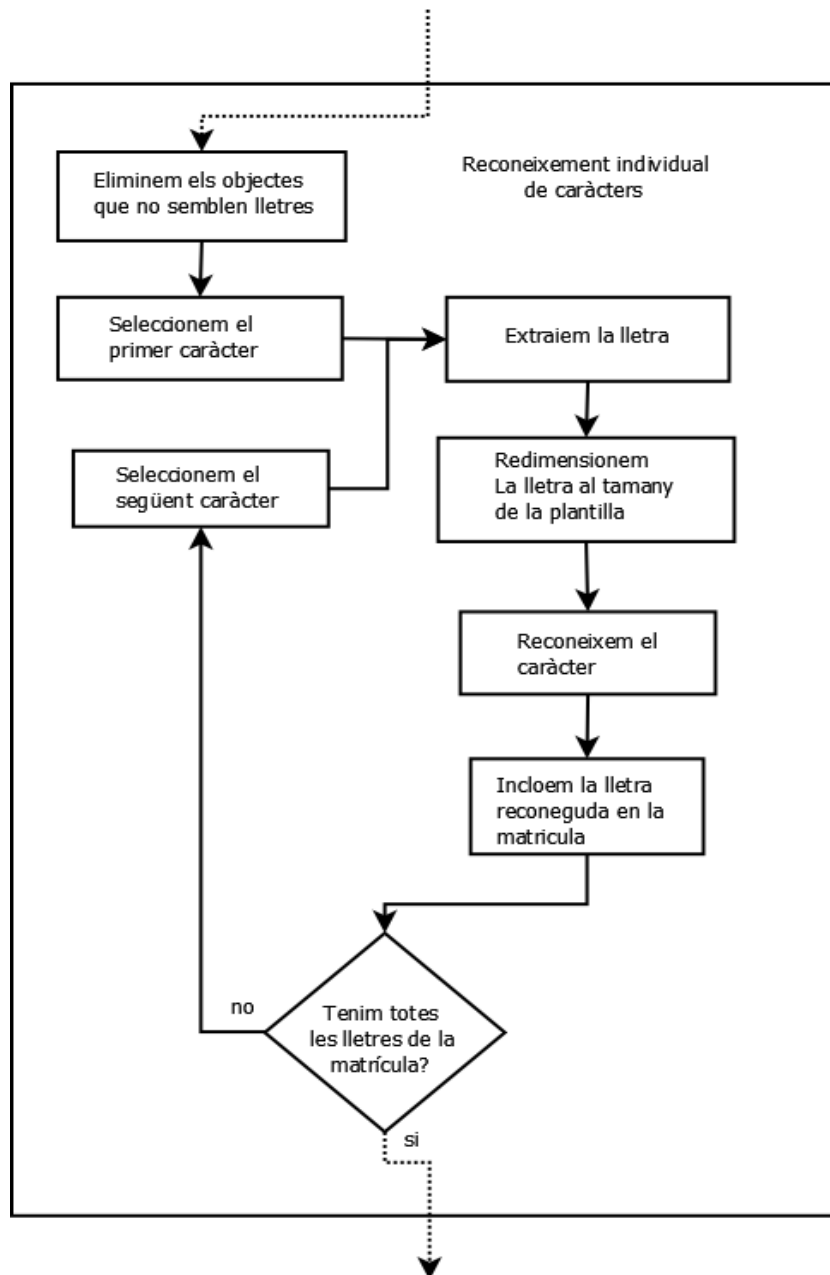


Fig. 16: Esquema del reconeixement individual de caràcters.

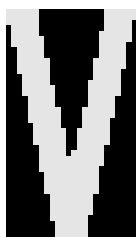


Fig. 17: Plantilla.



Fig. 18: Lletra de la matrícula.

Un com feta la correlació per tots els caràcters, escollim el màxim. Si aquest màxim no arriba al mínim exigít, el descartem. Si hi arriba, però no és un valor suficientment alt com per estar segur, utilitzarem dos processos més. Reduirem el caràcter a una simple línia connexa o esquelet, i mirarem els punts terminals del caràcter i els punts de creuament. Amb això, ho comparem amb el mateix procés aplicat a les plantilles. Si el nombre de punts coincideix en tots dos casos, podem seleccionar una lletra. Finalitzat això, es sotmet a consens l'elecció de la lletra, donant més pes a la solució de la correlació que als últims processos. Es pot veure un diagrama del procés de consens més detallat a la figura 36.



Fig. 19: Esquelet original i detectat.



Fig. 20: Punts extrems original i detectat.



Fig. 21: Punts de creuament original i detectat.

Amb aquest valor, determinem els casos amb més incertesa i així millorar el grau de precisió.

3.2.4. Validació de la resposta

Aquest procés es realitza dins el mateix reconeixement de caràcters, tant per limitar la quantitat d'operacions que es realitzaran, com per validar la resposta final del procés. Com s'ha descrit al principi d'aquest capítol, les matrícules tenen entre 7 o 8 lletres, segons el tipus de matrícula. Per matrícules noves, la xifra és de 7

caràcters, 4 números inicials i 3 lletres finals. Per matrícules antigues, la xifra varia entre 7 i 8 caràcters, amb 1 o 2 lletres inicials regionals, 4 números i 2 lletres finals. Amb aquestes regles, acotarem les comparacions de caràcters a lletres o números, segon calgui.

3.2.5. Tractament de les dades

Un cop acabat el procés de reconeixement de caràcters, mostrem el resultat obtingut i les dessem per al tractament posterior.

3.3. Dissenys previs

El disseny final de l'algorisme ha sigut conseqüència d'una evolució i integració de dissenys previs descartats. A continuació explicarem els dissenys previs de l'algorisme, els processos que varen sofrir i perquè els vam descartar.

El primer disseny es basava en fer operacions morfològiques sobre les imatges que obteníem per tal de descartar les possibles lletres. Fins a 13 passos s'involucraven per tal de extreure les característiques de les lletres (àrea que ocupa la lletra, excentricitat, relació d'aspecte, etc.), fins al final donar-li la regió de la matrícula al reconeixement de caràcters.

L'algorisme binaritzava la imatge de grisos directament, sense fer cap millora en la imatge, calculant que les lletres destaquessin sobre la matrícula cosa que no sempre s'aconseguia. Tot seguit, tractava les lletres com a objectes individuals i mirava característiques com la relació entre altura i amplada, que l'àrea de les lletres no superessin certs marges, etc. Finalment, calculava la orientació de la matrícula i girava la imatge segons això.

Aquesta implementació sofria de diversos problemes. Primerament, la imatge no es redimensionava, amb el qual les operacions morfològiques trigaven massa temps en calcular i modificar la imatge, i el temps d'execució era massa alt per als nostres requisits. Seguidament, aquest disseny era molt susceptible a l'escena, i si es trobava amb escenes complexes i molts contrastos, l'algorisme no detectava la matrícula, i les úniques solucions era ajustar els marges de reconeixement per que

detectés més objectes, cosa que dificulta la feina al reconeixement dels caràcters. Finalment, el mètode que empràvem per orientar la matrícula era molt deficient, ja que només girava la matrícula segons la orientació, no segons si la matrícula està orientada amb profunditat, cosa que donava més problemes a l'algorisme de reconeixement de caràcters.

Un avantatge que sí que tenia aquest primer disseny és el reconeixement de caràcters. Encara que existeixen mètodes de reconeixement de lletres més sofisticats, aquest algorisme ens era suficient per detectar els caràcters, encara que s'ha ampliat i millorat en la versió final.

Amb això, es va descartar completament aquest disseny i es va procedir a implementar un segon disseny. Aquest estava basat en un treball de Erik Bergenudd [2] on treballa amb imatges a molt baixa resolució i processos paral·lels per determinar la regió de la matrícula. Els processos s'encarregaven de mirar dues característiques de les matrícules: els canvis de color de blanc negre i viceversa, i que els caràcters negres estan junts entre ells. Aquests processos es comparaven per trobar la regió de la matrícula, processar-la per corregir la orientació i donar-la al algorisme de reconeixement de caràcters. Ben aviat es va detectar que aquest procés tenia un problema, i era que depenia enormement dels colors dels cotxes, on els cotxes foscos o negres no aconseguia detectar la matrícula. Com aquest error de disseny és difícil de solucionar, es va descartar aquesta via i es va iniciar el disseny final tal i com s'ha comentat en l'apartat 3.2.

El disseny final inclou parts de totes dues implementacions, tant el reconeixement de característiques de les lletres o de la matrícula del primer, com el sistema de correcció de la orientació del segon disseny.

3.4. Selecció de llenguatge

Aquí discutirem les alternatives de implementació, amb els avantatges i inconvenients, i ressenyes als llenguatges.

3.4.1. Implementació en C++

C++ és un llenguatge multiplataforma orientat a objectes, amb codi compilat. És possible controlar els recursos de la aplicació a baix nivell, i permet la programació completa de les funcions, optimitzant el rendiment.

La major dificultat de seleccionar aquest llenguatge és la inexistència de funcions de manipulació d'imatges integrades, i és molt costós programar un sistema de tractament d'imatges complet. Existeixen llibreries addicionals que implementen algunes de les funcions necessàries, però el tractament de les dades és molt complex i és insuficient per obtenir els resultats esperats.

3.4.2. Implementació en Java

Java és un llenguatge creat per Sun Microsystems, ara dins Oracle, de tipus orientat a objecte amb codi compilat en bytecode (per a una Màquina Virtual Java). Pren molta sintaxi de programació de C i C++, però conté un model orientat a objectes molt més simple, i la portabilitat del codi a altres sistemes operatius només depèn de la existència de la màquina virtual en aquell sistema.

Conté llibreries de manipulació d'imatges, però no es troben totes les funcions necessàries per dur a terme el projecte. La manipulació de dades és simple, però amb necessitat d'implementar codi per tenir funcionalitats addicionals, per tant és molt costós implementar el projecte completament.

3.4.3. Implementació en Matlab

Matlab és un llenguatge escrit en C i Java de la companyia MathWorks. És un llenguatge orientat específicament a càlcul matemàtic, amb paquets de funcions per el tractament de senyals i imatges. Gracies a això, el treball amb imatges és molt simple, i conté moltes de les funcions necessàries en el projecte.

Encara que principalment és un llenguatge interpretat, és possible empaquetar-ho en un fitxer autoextraïble per ser executat en un ambient d'execució de Matlab per tal de evitar la necessitat d'instal·lar el paquet complet del programa.

En contra, el llenguatge és multiplataforma però només està adaptat a les plataformes més extenses, és a dir, a Mac OS, Linux i Windows. Per tant, només podrem executar el nostre algorisme en aquests tres entorns, sense possibilitat d'executar-ho en altres sistemes operatius.

Hi ha diverses alternatives amb característiques similars a Matlab, la més coneguda i completa és Octave. Es va estudiar la implementació en aquest llenguatge, però degut a que la documentació era escassa i no té algunes de les eines de processament d'imatges que utilitzem en Matlab, varem afavorir aquest últim.

Tenint en compte només implementarem un prototipus de programa de reconeixement de matrícules, la manipulació de dades en Matlab és molt senzill i conté les funcions necessàries per el projecte, escollim aquest com a millor llenguatge. Les altres alternatives prendrien massa temps en implementar el disseny i seria molt propens a errors, essent molt difícil acabar en el temps estipulat.

4. Implementació

En aquesta part explicarem pas per pas l'algorisme implementat en Matlab. Per això dividirem cada part de l'algorisme per veure amb més detall els processos implicats. Totes aquelles paraules que estiguin escrites en *itàliques* són funcions pròpies de Matlab.

4.1. Adquisició de la imatge i processament previ

Ja que la idea és automatitzar el procés de captura d'imatges, considerem que aquesta part ha de ser el més simple possible. Per poder seguir els passos que componen l'algorisme, utilitzarem una imatge base (figura 22) i veurem tots els processos que necessitem per aconseguir la regió de la matrícula i el seu reconeixement.



Fig. 22: Imatge original.

Per tal captar les imatges posarem dos mètodes: captar una imatge amb una càmera digital compatible amb Matlab, o bé llegir les imatges d'un directori.

En el primer cas, comprovarem que hi hagi una càmera compatible instal·lada i detectada per Matlab. Un cop trobada, la inicialitzem i esperem a que l'usuari premi la tecla Enter per capturar la imatge, captarem la imatge i la retornem als

següents processos. En el segon cas, simplement llegirem la imatge d'un directori d'imatges.

Un cop obtinguda la convertirem a una imatge en escala de grisos, ja que la informació de color no ens fa falta i d'aquesta manera utilitzarem menys memòria per executar el programa.

El nostre algorisme serà independent de la resolució en la que obtenim la imatge, ja que modificarem la grandària d'aquesta. Si per exemple obtenim les imatges a una resolució de 8 MP (Megapíxels), és a dir, 3264 x 2448 píxels, o 7.990.272 píxels en total, reduïrem la imatge a una resolució de 800 x 600, o 480.000 píxels de manera estàndard. Amb aquest procés, aconseguirem reduir la quantitat de memòria i píxels amb que treballarem, i finalment reduir el temps d'execució sense reduir significativament la qualitat de la imatge (es poden veure els resultats de comparar diverses resolucions a la secció 5.1.3).



Fig. 23: Imatge redimensionada en escala de grisos.

També s'ha afegit la opció de treballar a altres resolucions natives, com ara 1024 x 768 píxels o 1600 x 1200 píxels, per tal de comparar els temps d'execució en aquests respecte a la resolució estàndard de 800 x 600.

Finalment, cridem la funció de extracció de la regió de matrícula amb la imatge que ens ha donat aquest procés.

4.2. Extracció de la regió de la matrícula

En aquesta part, extraïem la regió de la matrícula, fent els processos esmentats en l'apartat de Disseny 5.3.2.

4.2.1. Transformació de la imatge a binari

Tal i com s'ha comentat en la secció 5.3.2, millorarem els contrastos de la imatge i farem l'operador morfològic Bottom-Hat amb la funció *imbothat*.

Utilitzant un element estructurant en forma de matriu quadrada, *imbothat* ens deixa una imatge en grisos on les zones més petites negres les destaca en color més clar i les zones més grans de color fosc, ideal per detectar les lletres de la imatge. Es varen fer proves d'altres elements estructurants més sofisticats, com per exemple un disc, resultant que eren encara millors per detectar les formes de les lletres, però es va detectar que aquestes formes especials augmenten significativament el temps que triga en transformar la imatge, i amb la forma quadrada ens és suficient.



Fig. 14 Bottom-Hat

Seguidament, transformem aquesta imatge a binaria amb *im2bw*, per tal de seguir els processos de detecció. El llindar aplicat en aquest cas ha de ser un número comprès entre 0 i 1, on 0 és tot a blanc i 1 és tot a negre. Després de moltes modificacions en aquesta part, es va considerar que un llindar baix ajuda a treure

soroll de la imatge, i gràcies a les propietats de les matrícules les lletres queden molt contrastades del fons. Un cop binaritzada la imatge treballarem amb objectes, que és un conjunt de píxels blancs connectats amb un fons negre.



Fig. 25: Binarització.

Tot seguit trencarem tots aquells objectes que estiguin units per una quantitat molt baixa de píxels mitjançant *bwmorph*. Això ho farem per tal de separar caràcters que estan units per artefactes en la imatge (com poden ser cargols entre dues lletres, desperfectes petits en la matricula, etc).

4.2.2. Detecció de possibles caràcters

En aquest procés, apliquem una funció molt útil que utilitzarem en diverses parts del programa que ens dóna les estadístiques molt variades dels objectes en la imatge: *regionprops*.

Mitjançant aquesta funció, analitzem tots els objectes de la escena i eliminem el soroll de la binarització. Trobem les àrees més petites que un llindar predefinit per tal d'accelerar els processos posteriors i eliminant elements que no ens interessin de la escena. També utilitzarem aquesta funció més endavant en diversos punts per ajudar-nos a eliminar més elements. A continuació es mostra un exemple d'ús per extreure les àrees massa grans o petites:

```
lab=bwlabel(imchar); %etiquetem tots els objectes
props=regionprops(lab,'Area'); %obtenim les caracteristiques
id = find([props.Area]<MINAREA); %trobem les àrees més petites
imminarea = ismember(lab,id);
id = find([props.Area]>MAXAREA); %trobem les àrees més grans
immaxarea = ismember(lab,id);
imarea = imchar - imminarea - immaxarea; %treiem els objectes
```

La part més important és analitzar la proporció de les lletres, que és el que fem seguidament. Amb el mateix operand anterior, trobem els Bounding Box (apartat 3.2.2). Creem un altra imatge nova i mirem les relacions d'aspecte que tenen les caixes. Si excedeixen els paràmetres, no dibuixem la caixa en la nova imatge.

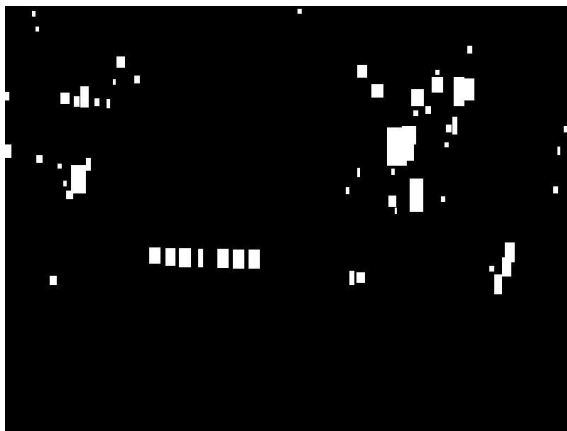


Fig. 26: Imatge amb Bounding Box.



Fig. 27: Imatge sense àrees petites i grans.

4.2.3. Detecció de línies i correcció de la orientació

Aquesta part del programa va ser implementada degut a la necessitat de reconèixer matrícules que no estiguin directament perpendiculars a la càmera. En entorns ideals, l'angle de la càmera és perpendicular o molt proper però en situacions menys controlades la inclinació de la càmera i l'angle fan que l'algorisme falli. Per corregir aquest tipus de situacions farem servir la transformada de Hough.

Per utilitzar aquest mètode primerament creem una nova imatge amb els contorns de les caixes. Per fer això en servirem de la funció *edge* de Matlab, utilitzant el mètode canny per detectar contorns.

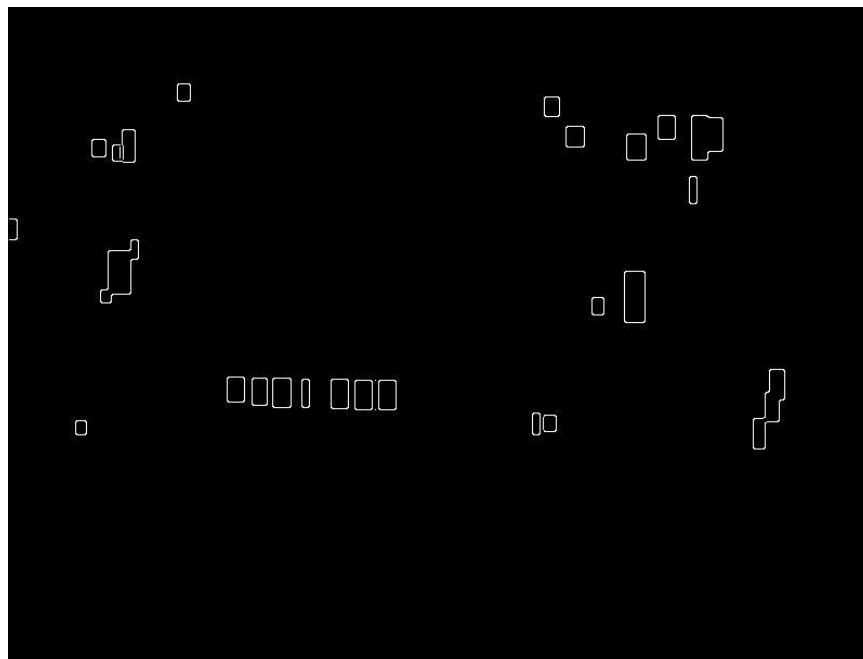


Fig. 28: Imatge amb els contorns de les caixes

Seguidament executarem la transformada de Hough, que ens guardarà una matriu acumulador amb totes les possibles rectes que passen per tots els punts de la imatge (veure figura 15).

Buscarem fins a 4 valors màxims d'aquest acumulador sempre que la inclinació de les rectes que ens dóna no sigui més gran que 45 graus respecte la horitzontal. També discriminem les rectes que creuin punts massa allunyats, ja que sabem que els caràcters es troben relativament junts.

Encara que aquest procés ens trobi 4 rectes diferents repartides per la imatge, els processos següents ens asseguraran que les caixes que hem seleccionat són matrícula.

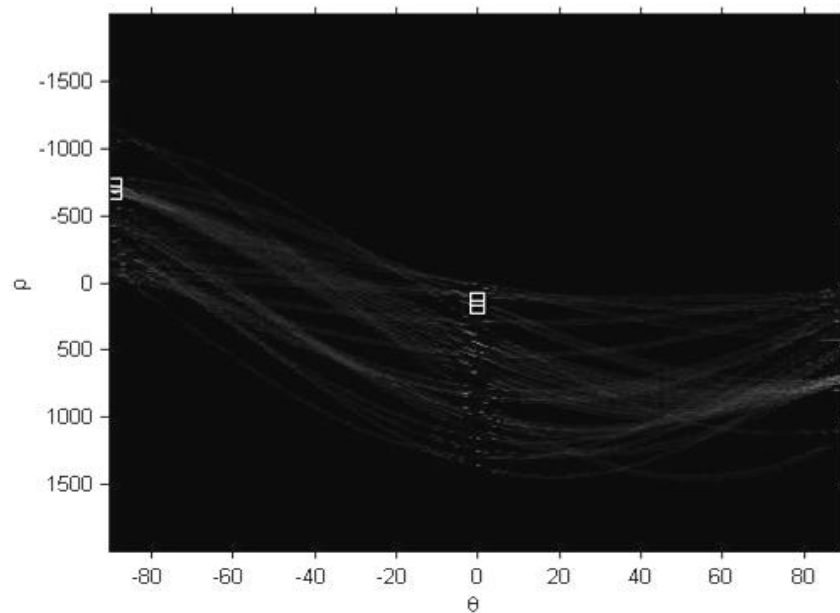


Fig. 15: Representació polar de la matriu acumuladora. L'eix vertical distancia rho amb l'origen, eix horitzontal angle theta de la recta amb els màxims marcats.

Com es pot apreciar en la anterior figura, Hough troba 4 màxims en la matriu acumulador, però només ens quedarem els dos màxims que estan compresos entre -45 i 45 graus de la horitzontal. El resultat final serà que tindrem dues rectes que tocan objectes, on aquests objectes sabem que formaran part de la matrícula.

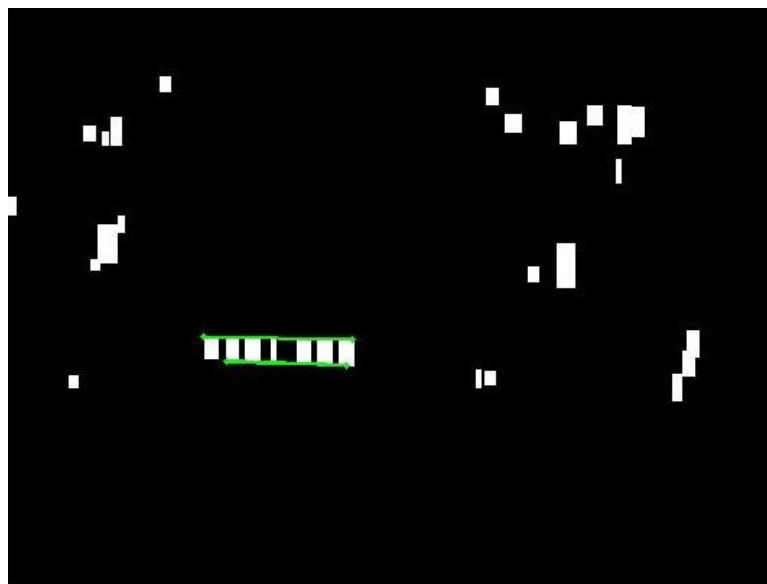


Fig. 30: Rectes de Hough (en verd) amb els Bounding Box.

Amb això, seleccionarem aquelles caixes que han format part de les línies obtingudes amb la transformada de Hough. Aquesta funció, *hough_bin_pixels* [10], ens marca els punts que han contribuït a la recta seleccionada. Un cop trobats aquests punts, etiquetem les caixes i les seleccionem per construir la imatge de la figura 31.

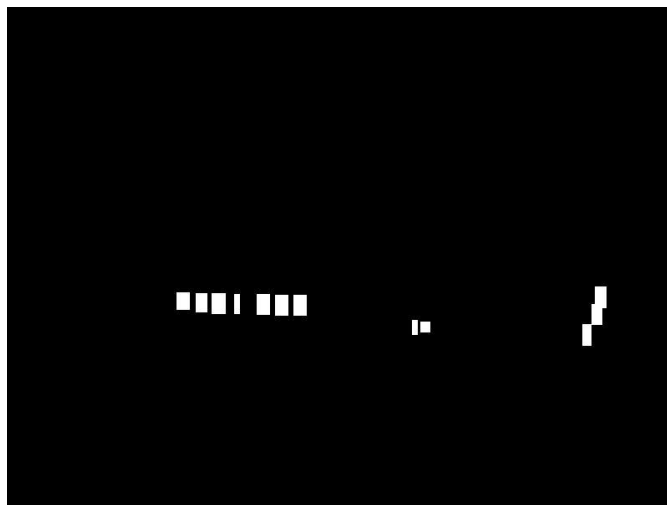


Fig. 31: Caixes seleccionades.

Aquestes rectes també tenen un altra funcionalitat en l'algorisme: ens permetran corregir la orientació i la profunditat de la matrícula. Mitjançant *maketform* crearem una matriu de transformació on corregirem la pendent de les rectes, i finalment orientarem la imatge amb la funció *imtransform*, deixant la imatge amb les lletres alineades per a les següents etapes.

4.2.4. Localització de possibles matrícules

En aquesta part, fem un closing per tal d'unir les caixes. L'element estructurant agafat és petit, per tal d'unir les caixes més properes i no fer-ho a altres més separades. Amb això aconseguim unir les lletres de la matrícula, deixant la regió molt ben definida, i així podrem discriminar millor la resta d'objectes de la escena.

L'última característica que mirarem és que les lletres, ara unides gràcies al closing tenen un aspecte semblant a un segment allargat. Per comprovar si efectivament això pot ser matrícula, i detectant la resta de soroll, farem servir la excentricitat de l'objecte mitjançant *regionprops*. L'excentricitat, mesurada de 0 a 1, ens mesurarà si la regió s'assembla a una circumferència (0) o si sembla més un segment (1). La regió de la matrícula ha de semblar més un segment, així que marcaren un valor proper a 1.

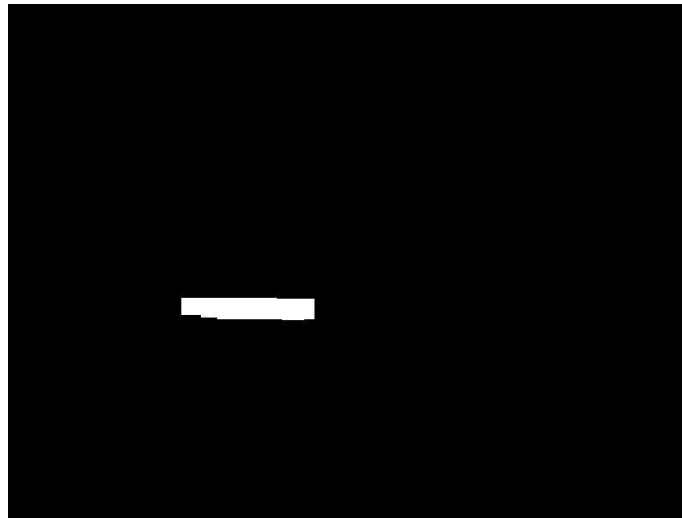


Fig. 32: Regió on es troba la matrícula.

Finalment, utilitzarem aquesta regió com a màscara per trobar els caràcters que componen la matrícula:



Fig. 33: Caràcters de la matrícula.

4.2.5. Extracció de la matrícula

Un cop localitzada la zona de la matrícula amb les lletres, retallem la secció i donem el resultat al procés de reconeixement de caràcters



Fig. 34: Matrícula final detectada.

4.3. Reconeixement individual de caràcters

A l'inici del procés de reconeixement, analitzem la regió per eliminar elements que no siguin lletres. En aquest cas, eliminem les àrees més petites a 30 píxels i mirem la proporció d'aspecte que tenen tal i com s'ha fet anteriorment.

Seguidament, fem el bucle principal del reconeixement de caràcters. Començant des de la última lletra fins a la primera, retallem cada caràcter i el redimensionem a una escala de 24 x 42 píxels, ja que les plantilles que utilitzarem després tenen aquestes dimensions. Com que aquesta part està molt unida a la Validació de la resposta, cada caràcter que trobem acota les possibilitats de les següents, és a dir, si és el cinquè caràcter per reconèixer només mirarem les xifres, ja que per qualsevol tipus de matrícula sempre és número. Això ens permetrà reduir el nombre de comparacions i accelerar encara més el procés.

```
while (n>=1)
    %Si ja tenim 7 o 8 caracters de la matricula, segons
    %quin tipus sigui, parem de processar
    if ((tipomatr==2) && (size(matr,2)==7)) ||...
        ((tipomatr==1) && size(matr,2)==8)
        break;
    end
    %Extraiem la lletra
    [hor,vert] = find(label==n);
    n1=ismember(label,n);
    n2=n1(min(hor):max(hor),min(vert):max(vert));
    %Redimensionem la lletra per ser idèntic a la plantilla
    imletra=imresize(n2,[42 24]);
    %Converteix la imatge en text i ho inclou en la llista
    %de caracters
    [letra numchar tipomatr] = ...
    leer_caracter(imletra,numchar,tipomatr);
    matr=[letra matr];
    n=n-1;
end
```

La part més important de l'algorisme és la comparació amb la plantilla. Les plantilles tenen unes dimensions de 24 x 42 píxels, i són extreptes de la tipografia que utilitzen les matrícules de cotxes. Gracies a que les matrícules utilitzen tipografies i mesures estàndard, aquest procés és relativament senzill i només en els casos extrems s'han d'analitzar altres característiques més complexes de les lletres.

Per tant, el que farem és crear una matriu de correlació, on indicarem amb un numero del 0 al 1 la 'semblança' amb cada plantilla. Quant més alt sigui aquesta xifra, més possibilitats de correspondència i ha amb aquesta lletra, i per tant, l'escollim com a candidat.



Fig. 35: Objectes redimensionats amb el caràcter reconegut.

El problema ve donat per la semblança entre diferents lletres, com per exemple la O i la D, o la P amb la R. Si l'apartat de la correcció de la perspectiva no ha sortit bé, és possible que agafin els candidats erronis.

En revisions properes a la versió final es va pal·liar aquest problema fent que l'algorisme analitzés dos paràmetres més i fes un consens entre ells per decidir el millor candidat.

El primer paràmetre s'extreu de reduir el caràcter a una línia connexa, deixant un esquelet de la lletra. El segon ens dóna informació sobre els punts de creuament de dos segments i els punts terminals de les lletres.

```
immatskel=bwmorph(imletra, 'skel', Inf);  
immatcruces= bwmorph(immatskel, 'branchpoints');  
immatext= bwmorph(immatskel, 'endpoints');
```

Un cop trobats aquests punts, dividirem la imatge en altres més petites i detectarem si el nombre de punts són iguals en totes les seccions. Aquelles lletres que s'assemblin més, seran el candidat que seleccionin.

Per acabar el procés, es sotmet a consens la solució per tal d'assegurar que la lletra és correcta, encara que es dóna més pes a la resposta donada per la correlació que no pas a les altres. Si la selecció de la plantilla falla i no arriba al mínim exigít, analitzarem únicament aquests dos paràmetres per trobar la lletra, y si coincideixen tots dos, donarem per bona la resposta. A continuació es mostra un diagrama del consens en les seves parts:

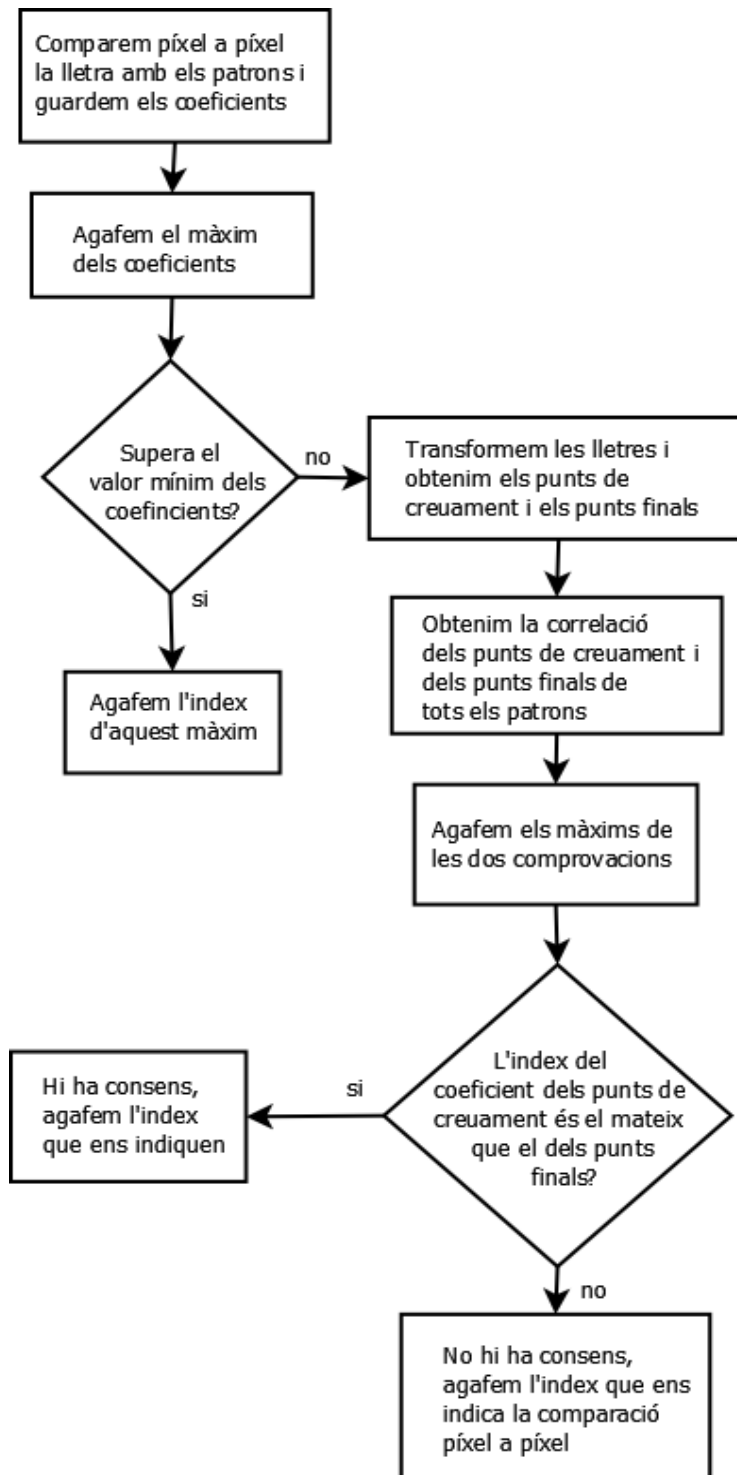


Fig. 36 Diagrama de consens.

Finalment, trobarem la lletra amb un selector i tornem a començar aquest procés fins que no quedin més lletres.

4.4. Validació de la resposta

Aquesta part treballa molt conjuntament amb la part de reconeixement de caràcters, tant per limitar el nombre d'operacions que s'han de fer com per determinar si la matricula és correcta.

Tenim dos principals vessants: la limitació de les respostes possibles de les lletres i la validació final.

La primera part, amb un comptador simple que controla el numero de caràcters ja detectats acota el bucle de comparació de les plantilles al rang que volem: si és lletra o si és numero. Això es determina segons les especificacions de les matrícules: Les dues xifres finals sempre compleixen que són lletres, la tercera xifra final és el punt crític de selecció si és una matricula nova o antiga, i finalment seleccionem la resta de possibilitats.

La segona part és una comparació simple, si la longitud de la matricula és de 7 o 8 caràcters segons el tipus de matrícula que hem detectat, ho donarà com a vàlid o no.

4.5. Tractament de les dades

Un cop acabat tot l'algorisme de detecció i reconeixement de matrícules ens retornarà la matricula. Amb aquest resultat ja podem manipular la dada segons les nostres necessitats: ho guardarem en un fitxer de text per poder tractar les dades en un futur, juntament amb l'hora que ha reconegut la matrícula.

5. Resultats

En aquest capítol exposarem els resultats de la nostra implementació. Per fer-ho s'han recopilat 350 imatges de cotxes, fotografiades des de diversos angles i obtenint tant la matrícula anterior com la posterior. També s'han aconseguit imatges en diverses llums ambientals i diverses hores del dia. Apart, s'han recopilat 86 imatges més de vehicles fotografiats de nit utilitzant el flaix de les càmeres o la llum ambiental.

En aquestes proves s'han utilitzat càmeres de diversos dispositius per tenir una gran varietat d'imatges diferents. Aquests dispositius són:

- Càmera digital Fujifilm Finepix J10, amb resolució màxima de 8,15 MP (3264 x 2448 píxels) amb longitud focal $f=6,2$ mm – 18,6 mm i zoom òptic 3x.
- Càmera integrada en Samsung Galaxy Note, amb resolució màxima de 8 MP, flaix LED i autofocus.
- Càmera integrada en Sony Xperia P, amb resolució màxima de 8 MP, flaix LED i autofocus, configurada a 2 MP.
- Càmera integrada en HTC Desire, amb resolució màxima de 5 MP.

Les màquines on s'ha executat l'algorisme són:

- **M1:** Portàtil Asus K55VM, CPU Intel i7 3610QM amb 4 nuclis i multifil, 3,1 GHz amb 6 MB de cache L2 i 6 GB de RAM DDR3, any 2012.
- **M2:** Ordinador de sobretaula amb CPU Intel i7 930 amb 4 nuclis i multifil, 2,8 GHz, 8MB cache L2 i 6 GB de RAM DDR3 a 1600 MHz, any 2010.

5.1. Precisió de l'algorisme

5.1.1. Precisió general

Aquesta prova mesura les matrícules correctes i incorrectes que aconsegueix l'algorisme. Per expressar els resultats utilitzarem una matriu de confusió, on definirem els possibles resultats de l'algorisme comparats amb la realitat:

- **True positive:** En la realitat hi ha una matrícula, i l'algorisme la detecta i reconeix bé. Són casos on el resultat és correcte.
- **True negative:** En la realitat no hi ha cap matrícula, i l'algorisme ens respon que no hi ha matrícula. No fem servir aquest estat ja que per les restriccions del sistema, sempre hi ha d'haver una matrícula en la imatge.
- **False positive:** En la realitat hi ha una matrícula, i l'algorisme la detecta, però no reconeix bé els caràcters donant-nos un resultat erroni.
- **False negative:** En la realitat hi ha matrícula, i l'algorisme no la detecta.

Un cop definides les possibilitats, la següent taula ens mostra les estadístiques de precisió de l'algorisme:

MATRIU DE CONFUSIÓ		Detectat	
		Positive	Negative
Realitat	True	293	0
	False	33	24

Taula 16: Matriu de confusió de l'algorisme.

Com es pot apreciar a la taula, l'algorisme té una precisió del 83,7%, mentre que la resta d'errors acumulen en el 16,3% dels resultats. Els errors més problemàtics són els false positive ja que l'algorisme ens dona una matrícula, encara que no podem saber si és errònia o no fins que no la comprovem manualment. Els false negative es poden solucionar prenent un altre imatge fins que reconegui la matrícula.

5.1.2. Precisió de l'algorisme en entorns de baixa il·luminació

A continuació mostrem els mateixos resultats per amb fotografies nocturnes. Encara que la quantitat d'imatges és més baixa, es pot veure una tendència dels resultats que dona l'algorisme en entorns de baixa il·luminació.

MATRIU DE CONFUSIÓ		Detectat	
		Positive	Negative
Realitat	True	64	0
	False	10	9

Taula 17: Matriu de confusió en entorns de baixa il·luminació.

En fotografies nocturnes, la precisió de l'algorisme baixa sensiblement, i depèn molt de que la imatge que s'ha pres estigui ben il·luminada, amb els caràcters ben contrastats. Això suposa un problema ja que les matrícules tenen una capa reflectant, que fa que els flaixos de les càmeres il·luminin la matrícula i a vegades saturen la imatge i resulta impossible llegir bé els caràcters.

5.1.3. Errors en la detecció de matrícules

A continuació focalitzarem en els errors que hem tingut de l'algorisme en fotografies diürnes, i analitzarem en quina de les etapes ha fallat. En aquest cas estudiarem els errors de l'etapa de detecció de matrícules i veure en quines etapes es concentren els errors de la detecció:

	Errors	Percentatge del total
Transformació a binari	5	20,8 %
Trobar els objectes que semblen lletra	11	45,9 %
Localitzar els objectes alineats i corregir la orientació	3	12,5 %
Localitzar les zones que semblen matrícula	5	20,8 %
Extracció de la matrícula	0	0 %

Taula 18: Distribució d'errors de la detecció de la matrícula.

Veiem que hi han errors en gairebé totes les etapes, però la majoria dels errors es deriven especialment en la segona etapa, trobar objectes que semblen lletra amb un 46% dels errors de la etapa de detecció. Per tant, deduïm que s'haurien de treballar més en aquesta etapa en particular per reduir al màxim possible aquests errors, ja sigui ajustant millor els paràmetre, o bé canviant tota aquesta part de l'algorisme per una més eficaç.

5.1.4. Errors en el reconeixement de caràcters

En aquest punt analitzarem els errors de l'algorisme en la part del reconeixement de caràcters.

Principalment, el reconeixement de caràcters falla particularment amb el soroll de la matrícula. Lletres tallades per la binarització, elements que podrien semblar lletres, etc. són la causa de molts dels falsos positius.

En els caràcters més problemàtiques trobem la D, la W, i l'1.

En el cas particular de la D trobem que la majoria dels errors es troben en com de bé s'ha fet el pas d'orientació de la imatge. Segons com estigui la pot reconèixer bé o confondre-la amb la lletra O.

El cas de la W és el més complicat, ja que els errors s'introdueixen des de el pas de la binarització, que talla la lletra i la separa, o bé amb el soroll de la matrícula la confon amb una I.

En el cas de les xifres, la més problemàtica amb diferència és l'1. Segons com estigui orientat el caràcter, el pot confondre amb un 7 o bé el 7 es confon amb l'1.



Fig. 37: Errors entre les lletres originals (a sobre) i lletres reconegudes (a sota).

5.2. Temps d'execució de l'algorisme

Ara executarem el programa per comprovar el rendiment de les diferents màquines descrites abans. La resolució estàndard del programa és 800 x 600 píxels, així que compararem de totes les imatges el rendiment mitjà d'aquesta resolució amb resolucions més altes per veure si canvia o empitjora el temps d'execució.

Començarem a mesurar el rendiment de l'algorisme de detecció de matrícules i l'acabarem un cop ens doni la resposta del reconeixement de caràcters, ja que la obtenció de la imatge (sigui del disc o de la càmera) i la escriptura en el disc de la matrícula depenen en gran mesura del sistema operatiu i com gestiona la entrada/sortida. Farem servir les funcions *tic* i *toc* de Matlab per mesurar aquest temps.

Les imatges incorrectes les desglossarem en els false negative i false positive, descrits en l'apartat 5.1.1. Això ho farem perquè l'execució s'atura abans en el cas dels falsos negatius, cosa que fa variar el temps d'execució.

M1:

		800 x 600 (s)	1024 x 768 (s)	1600 x 1200 (s)
True Positive		0,538	0,879	2,066
Incorrectes	False negative	0,626	0,940	2,076
	False positive	0,531	0,767	1,344

Taula 19: Temps d'execució en M1.

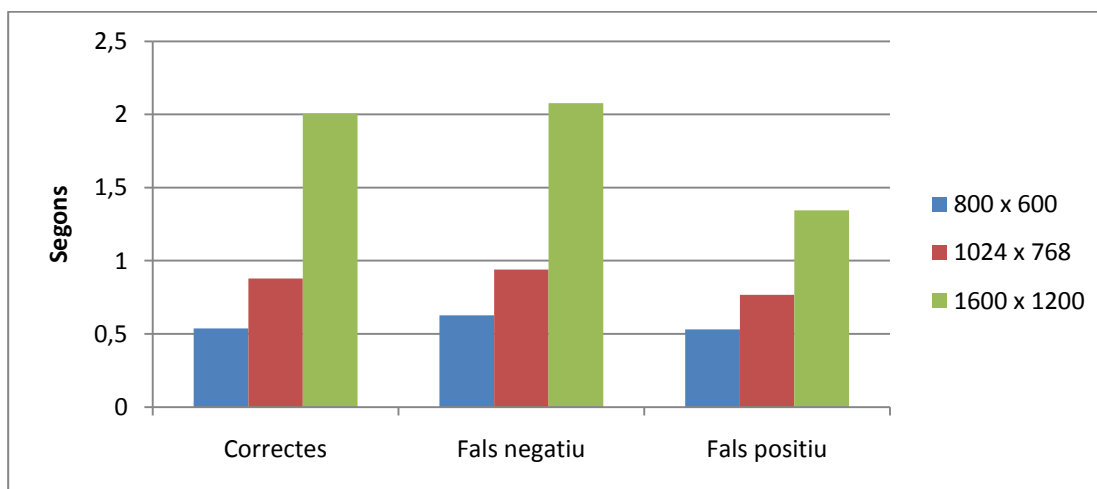


Fig. 38: Temps d'execució en M1.

M2:

		800 x 600 (s)	1024 x 768 (s)	1200 x 1600 (s)
True positive		0,783	1,220	2,802
Incorrectes	False negative	0,952	1,344	2,976
	False positive	0,770	1,109	1,778

Taula 20: Temps d'execució en M2.

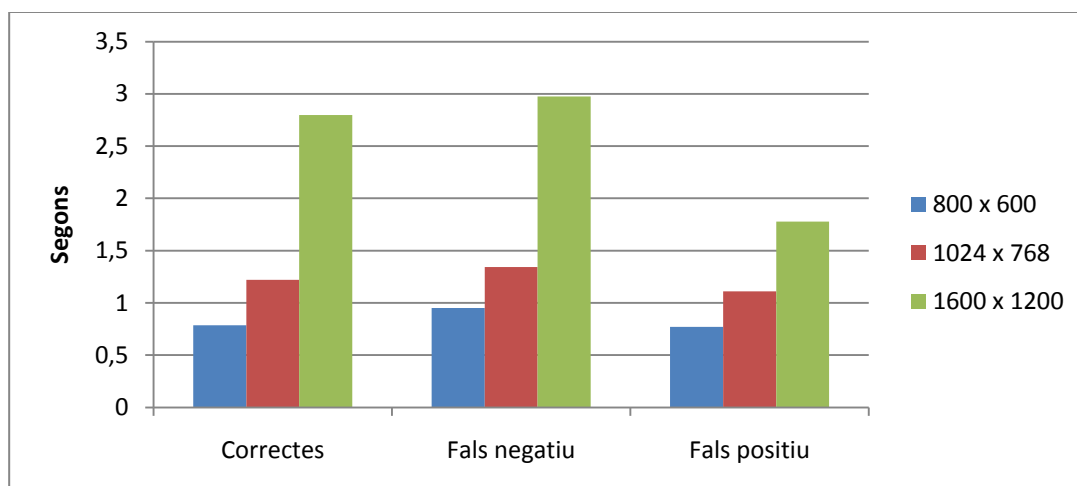


Fig. 39: Temps d'execució en M2.

Segons aquestes dades, observem que el temps d'execució augmenta amb la resolució, cosa que ja esperàvem. Processar una imatge amb resolució 1024 x 768 demana més recursos del sistema que la resolució estàndard, tant de memòria RAM com de temps del processador per donar la resposta. Amb això, la precisió de l'algorisme es manté i en certs casos baixa.

En la següent gràfica comparem els temps d'execució dels resultats true positive obtingut en totes dues màquines per veure quina tendència té l'algorisme.

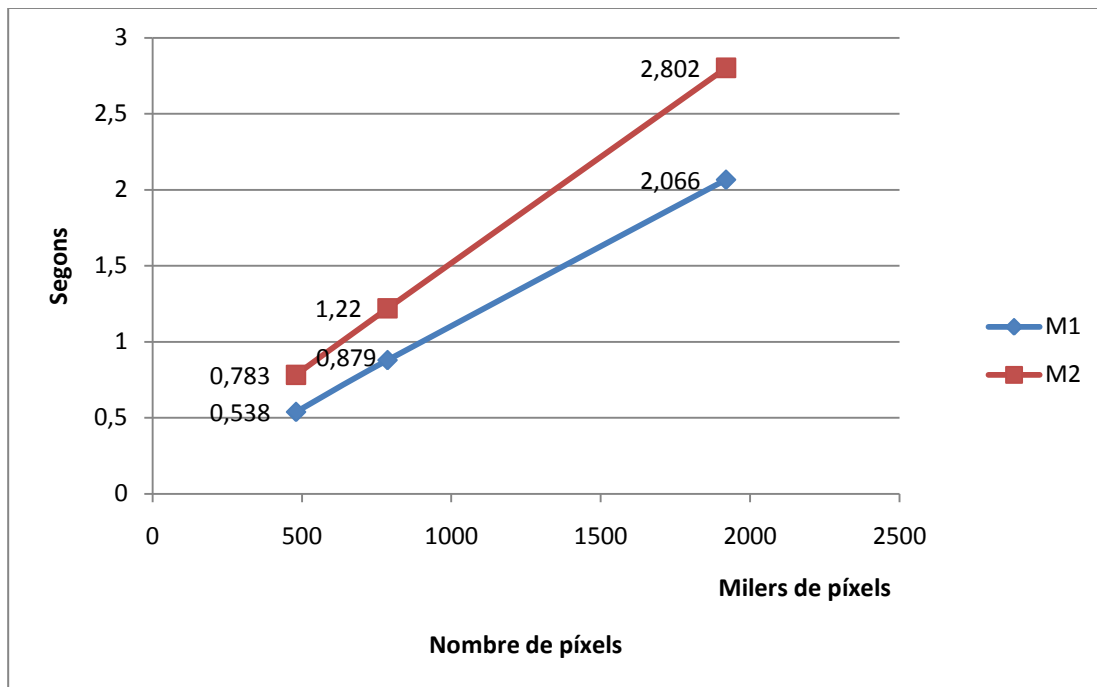


Fig. 40: Gràfica de temps segons resolució (mesurat en milers de píxels).

Les conclusions que es poden treure d'aquesta gràfica és que el temps d'execució de l'algorisme depèn en certa mesura del processador emprat. Amb una diferència de dos anys entre M1 i M2, tot i que tots dos són de gama similar, la diferència de temps d'execució entre ells arriba al 30%.

Ja que la gràfica anterior ens mostra el que sembla una recta, observarem la següent gràfica, que ens mostra el temps mitjà de processar un únic píxel per a cada resolució analitzada i entre les dos màquines.

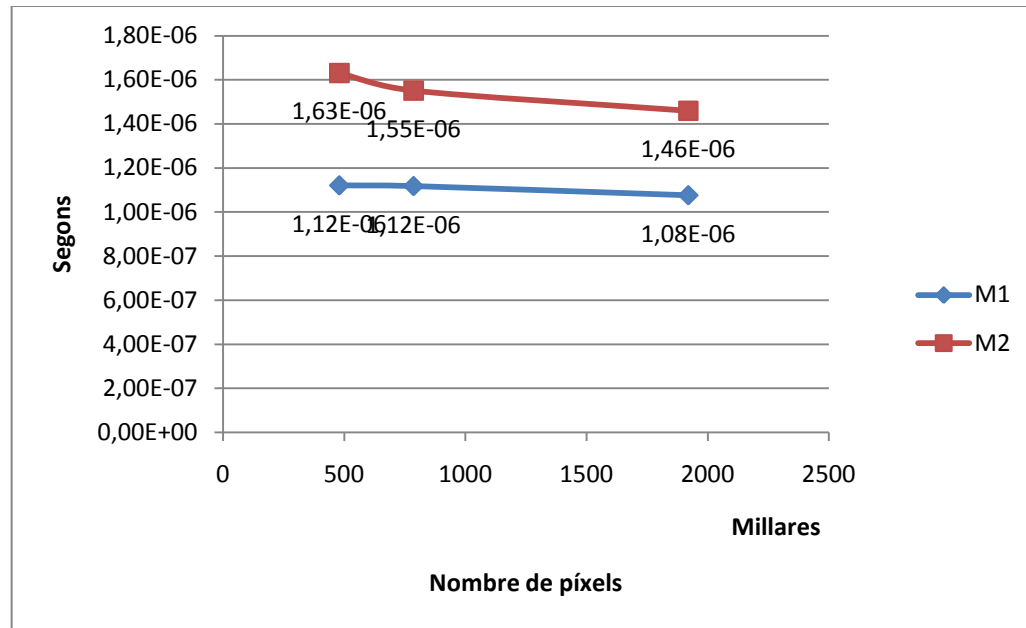


Fig. 41: Gràfica de coeficient de temps per píxel (mesurat en milers de píxels).

Encara que la gràfica decreix lleugerament en resolucions més altes, podem dir que el temps d'execució de l'algorisme és lineal, és a dir, que podem preveure, sabent el temps que triga en processar una imatge en una resolució, podem anticipar el que pot trigar en resolucions més elevades.

El fet que la gràfica decreixi pot estar atribuït al overhead o càrrega de treball de allocar l'espai de memòria, tractar les imatges o crear noves, etc. que estigui optimitzat de tal forma que sigui un temps constant, o bé a la poca precisió de la funció de comptador que utilitzem per fer les mesures.

5.3. Conclusió dels resultats

Els resultats obtinguts de l'algorisme han complert les expectatives i els objectius marcats del projecte. Aquests resultats s'han mesurat amb màquines que podem obtenir avui dia per preus econòmics, i hem utilitzat elements quotidians i de fàcil obtenció. Per tant, podem donar per satisfet un dels objectius principals: obtenir una precisió de l'algorisme suficient per ser eficaç i eficient.

L'altre objectiu principal que es proposàvem era resoldre aquest problema amb elements de baix cost. En aquest punt, i relacionat amb la precisió de l'algorisme,

podem dir que el temps d'execució és relativament baix i ha complert les expectatives.

6. Conclusions i treballs futurs

En aquesta secció farem un repàs dels resultats extrets i arribarem a les conclusions del projecte realitzat. També obrirem algunes línees de possible millora i ampliacions al projecte.

6.1. Desviació de la planificació

La planificació prevista del projecte no s'ha pogut complir. Algunes de les tasques previstes, sobretot les inicials, es varen sobreestimar i van prendre menys temps del planificat i altres tasques, sobretot la part de disseny i implementació, es van subestimar i varen prendre molt més temps del previst, cosa que va afectar a tot el projecte i no es va poder finalitzar i presentar la data prevista. Un cop sobrepassat la data de finalització, i per motius aliens, es va paraitzar el projecte fins al juliol del 2012, on es va poder reprendre, acabar i millorar la implementació i sobretot dedicar més temps a la memòria del projecte.

A continuació es pot veure la comparació de la planificació final i la prevista. Cal tenir en compte que la dedicació d'hores al projecte no ha variat massa durant el projecte, essent aquest una mitjana de 4 hores diàries. El càlcul d'hores es d'aproximadament 400 hores de dedicació.

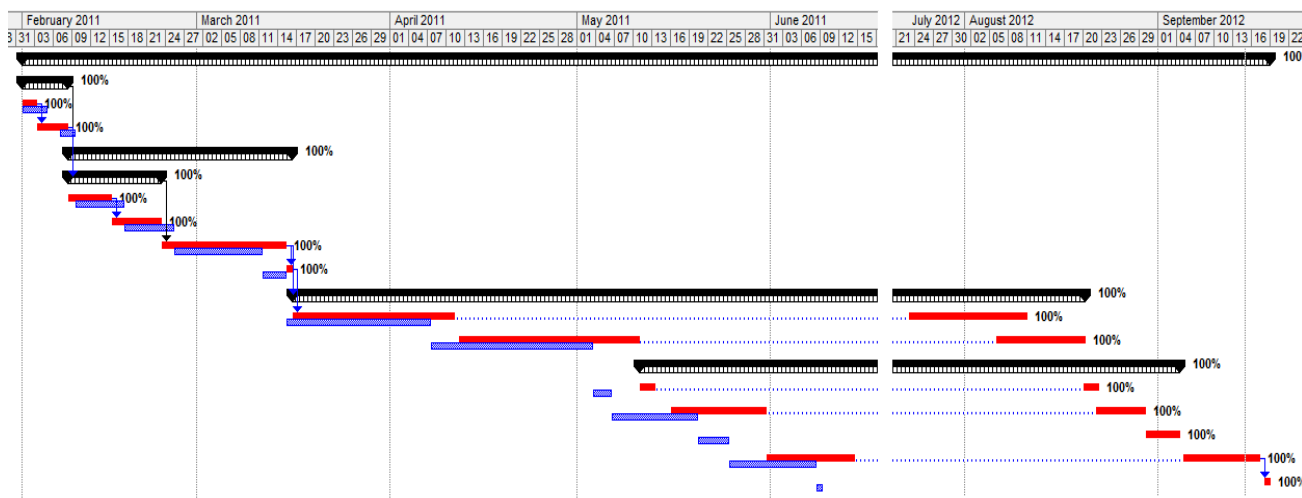


Fig. 42: Planificació final (vermell) i planificació prevista (blau).

6.2. Conclusions

Al llarg del projecte hem estudiat diverses alternatives dels mètodes que s'utilitzen per detectar matrícules de cotxes, hem analitzat els seus punts febles, implementat i extret resultats d'un nou algorisme per detectar matrícules amb sistemes de baix cost. Per fer això, hem revisat exhaustivament la documentació disponible dels mètodes de detecció de matrícula i hem escollit un apropament relativament senzill i ràpid d'executar, basant-se en operacions morfològiques. Per validar el nostre algorisme, també s'ha recopilat una gran quantitat de fotografies de matrícules de cotxe, i l'hem testejat per veure la seva eficàcia.

Tenint en compte tot això, el projecte final de carrera ha sigut un desafiament molt gran per mi, des de el mateix principi de l'estudi de l'estat de l'art on trobem mètodes que funcionen bé i altres que no tant, passant per la organització de les tasques que es necessitaven fins a la implementació del programa.

Tenint en compte la naturalesa d'aquest projecte, era clar que s'hauria de revisar el disseny de algorisme més d'una vegada, escrivint parts de codi i provant-ho cada cop, pas per pas fins a tenir un programa complet. També s'ha de tenir present que calibrar cada pas i cada procés era una tasca de prova i error, i per tant moltes hores s'han dedicat en aquests esforços.

Com a part positiva d'aquest projecte em trobo que m'ha ajudat a aprendre a organitzar totes les tasques d'un projecte de característiques similars a les que trobaríem en entorns laborals reals, on hem d'identificar el problema que existeix, analitzar les possibles solucions i alternatives i un cop escollit, dissenyar i implementar l'algorisme que ens ho solucioni. Això mateix no ho podem trobar en les pràctiques que hem dut a terme al llarg de la carrera, ja que eren molt pautades i de curta durada.

Escollir Matlab per el projecte era una petita dificultat per començar. Gairebé no ho hem utilitzat en la carrera, i la manera de treballar i programar difereix d'altres llenguatges. Té molta semblança a altres llenguatges estudiats, però amb peculiaritats que he necessitat d'aprendre per poder continuar. D'altra banda Matlab ha facilitat molt la feina en altres camps, com per exemple en comprovar els temps d'execució, o fins i tot un cop ben configurat ens ha permès compilar el programa en un executable.

6.3. Ampliacions

Com a ampliacions a aquest projecte, es poden obrir diversos camps de millora i estudi per millorar la eficàcia i el rendiment de l'algorisme.

Primerament, en referència al reconeixement de caràcters, es podria investigar la possibilitat d'incloure algorismes més eficaços, com ara sistemes automàtics d'aprenentatge on el mateix programa guardés imatges reals de matrícules per utilitzar-ho com a base de dades per contrastar les lletres trobades. Molt possiblement aquesta implementació augmentés el temps d'execució de l'algorisme, però la quantitat d'èxits augmentaria significativament. També es poden utilitzar sistemes més complexos de reconeixement de caràcters, com ara bé xarxes neuronals on els processos es poden paral·lelitzar i reduir el temps d'execució.

D'altra banda, i segons els resultats obtinguts a l'apartat 5.1.3, és millorar la part de detecció de matrícules, en especial el pas de trobar objectes que semblen lletra, ja que era més problemàtic. També es pot incloure un sistema de detecció de matrícules en moviment, amb un vídeo obtingut per una càmera i que analitzi constantment les matrícules que apareixen en l'escena i s'automatitzi el sistema de detecció i reconeixement.

Una última possible ampliació és incloure aquest programa dins d'un software de gestió que treballi amb les matrícules detectades, manipuli les dades i controli l'algorisme per tal de facilitar l'accés a un recinte i comprovi el temps d'estància al recinte, o detectar el pas de cotxes dins d'una zona de peatge.

7. Bibliografia

7.1. Publicacions consultades

- [1] Baldrich, Ramon. (2009). "Segmentació". A: Visió Artificial. Sabadell: Escola Universitària d'Informàtica.
- [2] Bergenudd, Erik. (2006, Desembre). "Low-Cost Real-Time License Plate Recognition for a Vehicle PC", Projecte de Máster presentat a KTH Electrical Engineering, Suècia.
- [3] Bharatkumar Gohil, Naikur. (2010). "Car License Plate Detection", Projecte de Máster presentat a California State University, Sacramento.
- [4] Martinsky, Ondrej. (2007). "ALGORITHMIC AND MATHEMATICAL PRINCIPLES OF AUTOMATIC NUMBER PLATE RECOGNITION SYSTEMS". B.Sc. Thesis de Brno University of Technology.
- [5] Shyang-Lig, Chang. (2004, Març). "Automatic License Plate Recognition". Intelligent Transportation Systems, IEEE Transactions on. (volum 5, núm. 1, p.42-53).
- [6] AEPD. Matrículas de vehículos y concepto de dato de carácter personal. Informe 425/2006. (2006).
- [7] Llei Orgànica 15/1999, de 13 de desembre, de Protecció de Dades de Caràcter Personal.
- [8] Orden de 15 de septiembre de 2000 por la que se modifica el reglamento general de vehículos (R.D. 2822/1998, de 23 de Diciembre).

7.2. Referències electròniques consultades

- [9] Automatic Number Plate Recognition, CARMEN FreeFlow.
<http://www.anpr.net/anpr_09/anpr_freeflow.html>. [Últim accés:
17/09/2012].
- [10] Steve Eddins, funció Hough_bin_pixels, versió del 05/09/2006.
< <http://www.mathworks.com/matlabcentral/fileexchange/12142-hough-accumulator-bin-pixels>>. [Últim accés: 17/09/2012].
- [11] Wikipedia.org. Aprenentatge supervisat.
<http://ca.wikipedia.org/wiki/Aprenentatge_supervisat#Resultats_de_les_prediccions>.
[Últim accés: 17/09/2012].
- [11] Matlab Documentation Center.
<<http://www.mathworks.es/es/help/documentation-center.html>>. [Últim
accés: 17/09/2012].
- [12] Stackoverflow.
<<http://stackoverflow.com/>>. [Últim accés: 17/09/2012].
- [13] Wikipedia.org. Car Plate Recognition.
<http://en.wikipedia.org/wiki/Car_plate_recognition>. [Últim accés:
17/09/2012].

Signat: Alberto Valero Moreno

Sabadell, Setembre de 2012